Mitigating the Performance Degradation caused by Execution Units Contention via Instruction-Aware Mapping

Matheus S. Serpa[1], Eduardo H. M. Cruz[2], Matthias Diener[3], Antonio C. S. Beck[1], Philippe O. A. Navaux[1]
[1]Institute of Informatics, Federal University of Rio Grande do Sul, UFRGS, Brazil
[2]Federal Institute of Paraná, IFPR, Brazil
[3]University of Illinois at Urbana-Champaign, USA
{msserpa, caco, navaux}@inf.ufrgs.br, eduardo.cruz@ifpr.edu.br, mdiener@illinois.edu

***Abstract*—Parallel applications running on Simultaneous Multithreading (SMT) processors compete for execution units. This issue is aggravated when threads execute similar instructions such as branch, floating-point, integer, load, and store. In this case, the same kind of instruction is dispatched for execution, leading to sequential execution due to the contention of execution units, resulting in a performance loss. State-of-art focuses mostly on proposing techniques that reduce memory contention on multiprogrammed workloads, which share fewer resources and are mostly heterogeneous. This work proposes a transparent method that automatically maps threads of multiple parallel applications on SMT processors. We focused on improving performance by mitigating the contention on execution units when running parallel applications. To achieve that, we map threads that stress the same execution units to different cores. In order to evaluate our proposal, we use a microbenchmark and the OpenMP version of NAS Parallel Benchmarks (NPB). Results show performance gains of 29.1% and 17.4%, on average, compared to the native scheduler of the operating system and a Round-robin mapping, respectively.**

***Index Terms*—SMT processors, Performance degradation, Resource sharing, Execution unit contention.**

ACKNOWLEDGMENT

This work has been partially supported by Petrobras (2016/00133-9, 2018/00263-5) and Green Cloud project (2016/2551-0000 488-9), from FAPERGS and CNPq Brazil, program PRONEX 12/2014. This study was financed in part by the Coordenao de Aperfeioamento de Pessoal de Nvel Superior - Brasil (CAPES) - Finance Code 001.

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

DISCLAIMER

This abstract describes the paper entitled **Mitigating Execution Unit Contention in Parallel Applications using Instruction-Aware Mapping**, from the same authors submitted to the 22nd IEEE International Conference on High Performance Computing and Communications (HPCC 2020).