DEFESA DE TESE DE DOUTORADO

Programa de Pós-Graduação em Ciência da Computação

Subutai: Distributed Synchronization Primitives for Legacy and Novel Parallel Applications

ALUNO: Rodrigo Cadore Cataldo
ORIENTADOR: Dr. César Augusto Missio Marcon
COORIENTADOR: Dr. Jean-Philippe Diguet
COORIENTADOR: Dr. Kevin J.M. Martin
BANCA EXAMINADORA: Dr. Márcio Eduardo Kreutz (PPgSC/UFRN), Dr. Avelino Francisco Zorzo (PPGCC/PUCRS), Dr. Eduardo Augusto Bezerra (PPGEEL/UFSC)
DATA: 16 de dezembro de 2019
LOCAL: Prédio 32, sala 506.2
HORÁRIO: 09:00

RESUMO:
Parallel applications are essential for efficiently using the computational power of a MultiProcessor System-on-Chip (MPSoC). Unfortunately, these applications do not scale effortlessly with the number of cores because of synchronization operations that take away valuable computational time and restrict the parallelization gains. Multiple solutions in research and industry have been proposed to tackle the synchronization bottleneck: they range from software-based, hardware-based, to mixed solutions. However, either they restrict the application to a subset of synchronization primitives, require refactoring the source code of it, or both.
We address these limitations and introduce Subutai, a hardware/software architecture designed to distribute essential data synchronization mechanisms over the Network-on-Chip (NoC). Subutai is comprised of novel hardware specialized in accelerating synchronization operations, a small private memory (1KiB) required for recording events, an operating system driver, and a user space custom library that supports legacy and novel parallel applications. For this work, we target the POSIX Threads (PThreads) library as it is widely used as a synchronization library, and internally by other libraries such as Open MultiProcessing (OpenMP) and Threading Building Blocks (TBB). Additionally, we provide extensions to Subutai intended to further accelerate parallel applications in two scenarios: (i) multiple applications running in a highly-contended scheduling scenario; and (ii) remove the access serialization to condition variables in PThreads. Experimental results with four applications from the PARSEC benchmark running on a 64-core MPSoC show an average application speedup of 1.57x compared to the same architecture running legacy software solutions. The same applications are further sped up to 5% using our proposed Critical Section-aware (CSA) scheduling policy compared to a baseline Round-Robin (RR) scheduler without any changes to the application source code.