Conference in Numerical Analysis 2014 (NumAn 2014)
September 2-5, 2014
Chania,Greece

Application of a hybrid parallel Monte Carlo PDE Solver on rectangular multi-domains[1]

Panayiotis Alefragis[a], Alexandros Spyrou[b], Spiridon Likothanassis[b]
[a]Computer and Informatics Engineering Dept., Technologigal Educational Institute of Western Greece,
Antirrio, Nafpaktos, Greece
[b]Computer Engineering and Informatics Department, University of Patras,
Rio, Patras, Greece
alefrag@teimes.gr,kspyrou,likothan@ceid.upatras.gr

Abstract

In this paper sequential performance analysis and the parallel performance achieved using a hybrid parallel implementation of a PDE solver is presented. In the current implementation the distributed memory programming model using MPI is combined with the shared memory programming model using POSIX threads. By using the hybrid mode-programming model we try to take advantage of the benefits of shared and distributed memory models without paying the disadvantaged of each model by utilizing shared memory between cores of each SMP node to minimize communication overhead and gaining scalability by harnessing the performance of multiple clustered servers.

This paper analyzes the sequential performance of a PDE solver that is applied on Elliptic PDEs (and Poisson in particular) on rectangular multi-domains in both two and three dimensions. The sequential analysis revealed code kernels that were highly coherent and could not be efficiently parallelized in a distributed environment, while they could benefit from a shared memory environment. These kernels can only be orchestrated at a higher level in a distributed environment by relaxing synchronization between cooperating servers. Based on the results of the analysis, a hybrid parallelization approach combining MPI tasks with POSIX threads is proposed in order to exploit the different levels of parallelism of actual multicore architectures. The paper presents implementation details and extensions to the sequential algorithm in order to allow efficient implementation in a distributed memory environment. Comparison between the performance of sequential, distributed memory, shared memory and the hybrid impementations are presented. The hybrid implementation manages to demonstrate significant speedup while maintaining the quality of the solution on sample input datasets. Finally, insight for further extensions to the proposed algorithm are presented.

Key words: Hybrid Parallel Programming, MPI, POSIX Threads, Monte Carlo PDE solver.

[1]The present research work has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program *Education and Lifelong Learning* of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS. Investing in knowledge society through the European Social Fund.