Hessisches Kompetenzzentrum
für Hochleistungsrechnen

Performance Analysis of BoSSS Solver Package (Bachelor Thesis)



Project Manager
Jan-Patrick Lehr

Researchers
Dr.-Ing. Florian Kummer, Dennis Krause and Verena Sieburger

Principal Investigator
Prof. Dr. Christian Bischof

Project Term
2018 - 2019

Project Areas
Computer Science

Clusters
Lichtenberg Cluster Darmstadt

Software
BoSSS

Additional Software
Intel vTune Amplifier, HPCToolkit, mono, perf, MUMPS, MKL

Institute
Department of Computer Science

University
Technische Universität Darmstadt

Introduction

The Bounded Support Spectral Solver (BoSSS) is developed as a flexible solver package to enable research in (mostly) fluid dynamics. Although the performance characteristics are known at a high- level, the actual behavioral properties of the compute kernels have not yet been investigated. In this project, we approach the BoSSS solver library with a structured, performance engineering workflow. Initially, the investigation is limited to the already identified, most important regions of the code. Our goal is to understand the behavior of these regions, the main influencing factors for the observed behavior and develop potential improvements to speed up the computation. As the solver is implemented in C# our research also briefly evaluates the availability and applicability of performance analysis tools for managed languages and compare that to established performance profilers from the HPC community.

Methods

To approach the performance analysis, we used a structured approach as proposed in [1]. We apply well-known HPC performance analysis tools, although the target application is written in C#, to investigate their usefulness in such an environment. Both tools, HPC Toolkit and Intel vTune, are sampling-based profilers that allow to identify code hot-spots and capture hardware metrics, such as the number of floating point operations or loads and store.

Results

The application of the typical HPC performance analysis tools did reveal insight into the behavior of the native code parts. However, the managed-code parts, i.e., the C# regions were not sufficiently covered. We applied manual instrumentation to capture more information about the runtime behvaior of the C code parts. We found that the application of the MUMPS solver with its current settings is the major limitation of the BoSSS application in our test case. It was due to using the underlying BLAS library, i.e., Intel's MKL, in an OpenMP-parallelized version, while the MUMPS solver itself was not built with threading support. This resulted in an non-optimal use of available node-level parallelism and contention on OpenMP synchronization primitives. The findings are reasonably specific for the scenario investigated, and, despite the synchronization bottleneck, showed that the large test-cases can benefit from two-way thread parallelism in the BLAS library to result in a speed-up of up to factor 1.8x.

Discussion

Our findings were reported back to the developers and can be used to optimize the identified regions. Furthermore, they can be used to implement performance stewardship methods, such that the code automatically generates feedback to the programmer about it's efficiency, given the current parameter settings.

Publications

Sieburger, Verena. "Performance Analysis of the Bounded Support Spectral Solver" (2019).

Reference

[1] Iwainsky, Christian et al. "Enhancing brainware productivity through a performance tuning workflow." In: Euro-Par 2011: Parallel Processing Workshops (2011), 198–207, Springer.
https://doi.org/10.1007/978-3-642-29740-3_23

Last Update: 2021-08-19 13:50