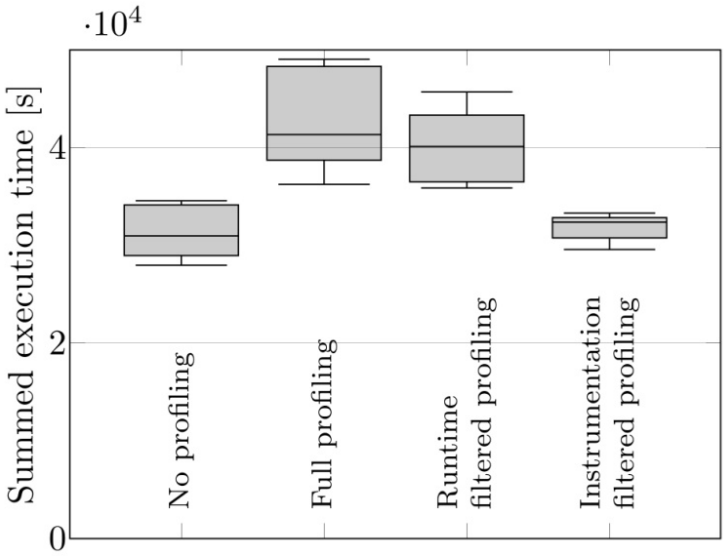
Hessisches Kompetenzzentrum
für Hochleistungsrechnen

Practical Empirical Performance Modeling for CFD Applications using Extra-P



Project Manager
Lukas Rothenberger

Researchers
Alexander Geiß and Gustavo de Morais

Principal Investigator
Prof. Dr. Felix Wolf

Project Term
2024 - 2025

Clusters
Lichtenberg II Cluster Darmstadt

Software
OpenFOAM

Institute
Parallel Programming

University
Technische Universität Darmstadt

Introduction

HPC applications can contain implicit performance bottlenecks, for example, due to caching or synchronization effects, which only manifest themselves once the scale of the HPC system reaches a certain magnitude. While mathematical performance models of an application can be used to identify such bottlenecks before scaling the system, their creation is a challenging task due to the size and complexity of modern HPC applications such as configurable CFD solvers. The laborious, manual, analytical creation of performance models is not practical, while the automatic generation of performance models using tools like Extra-P on the basis of empirical Score-P profiling has its own challenges. Especially, profiling a full application can incur large memory- and runtime overheads, which negatively influence the accuracy of the created models. In an effort to mitigate both the introduced overheads and inaccuracies in the models, we developed a simple, cheap, and generally applicable approach for Score-P filter file creation based on the results of a single profiling run. By example of the lid-driven cavity flow benchmark contained in the OpenFOAM HPC Benchmark Suite, we evaluated the accuracy of the created performance models in a strong-scaling scenario, and thereby established an example how to approach the empirical performance modeling of strong-scaling scenarios for large, complex applications using Extra-P.

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg

Methods

Inspired by prior research on source code hotspot detection, the automatically created filter allows the profiling of functions, which contribute significantly to the modeling due to either a large individual or accumulated runtime. Excluded from the profiling are such functions, which are frequently visited but have short runtimes, as they introduce a comparatively large overhead without contributing significantly to the actual runtime of the program and thus decrease the model quality. As an additional contribution, we conducted a case study on a strong scaling application using Extra-P. While Extra-P is primarily designed for weak scaling, our study extends its application to strong-scaling scenarios by modeling computational effort in core seconds rather than the observable execution time. As a result, the relationship between effort and the number of processors in the system would be represented by a constant function in a perfect strong-scaling scenario.

Results

Our experiments have shown that the proposed, simple approach for automatic filter file creation can lead to a strong reduction of the introduced profiling overhead—from a factor of 1.34 for the full instrumentation to a factor of merely 1.05 when applying the created filter file during the instrumentation process (see Fig. 1). With regard to the created performance models we were able to observe an evident similarity between the models created using measurements on up to 6 nodes of the Lichtenberg 2 cluster (see Fig. 2, left) and the reference models created using up to 14 nodes (see Fig. 2, right), thus showing the ability to extrapolate the runtime behavior of the considered functions when increasing the system’s size and identify potential bottlenecks (e.g., MPI Allreduce in gSumProd in Fig. 2) within a reasonable projection distance.

Discussion

While the project showed promising results regarding the filter file and strong-scaling performance model generation, the considered lid-driven cavity flow benchmark is arguably one of the simplest and best-studied cases from the OpenFOAM HPC Benchmark Suite. As a result, the immediate findings regarding the scaling behavior and potential bottlenecks are not particularly interesting but align with the application’s previously established analytical performance models, demonstrating the accuracy of the empirical performance models. For this reason, we argue that, although more complex code configurations and different software suites remain to be tested, the proposed approach for empirical performance modeling of complex code is a viable way to identify potential performance bottlenecks in strong-scaling scenarios while reducing the manual as well as computational effort significantly.

Figures

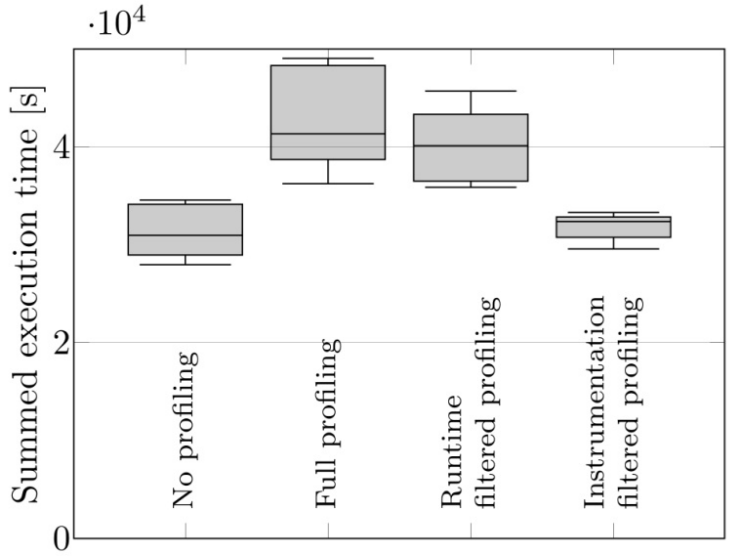


Figure 1: Effects of different filtering schemes on the required resources during profiling.

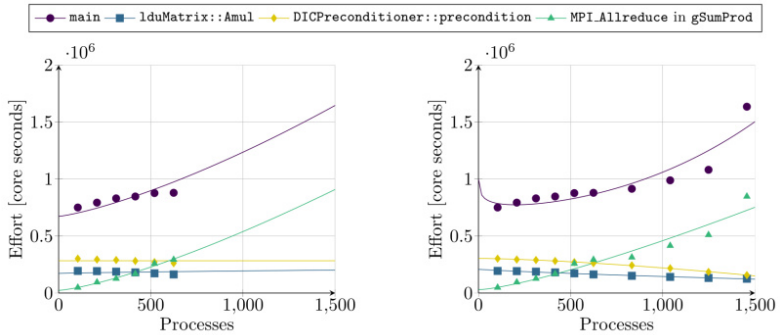


Figure 2: Effort performance models for the lid-driven cavity flow example of OpenFOAM, created using up to 6 nodes of the Lichtenberg 2 cluster (left) for exemplary model generation and up to 14 nodes (right) for validation purposes.

Publications

de Morais, G.: "Performance Modeling for CFD Applications", 35th International Conference on Parallel Computational Fluid Dynamics (ParCFD2024), Bonn, Germany, Sep. 02-04, 2024

de Morais, G.: "Performance Modeling for CFD Applications", 2nd NHR Conference, Darmstadt, Germany, Sep. 09-10, 2024

de Morais, G.; Geiß, A.: "Performance Modeling for CFD Applications", NHR4CES Community Workshop, Online, Jun. 13-14, 2024

Rothenberger, L.; de Morais, G.; Geiß, A.; Wolf, F.: "Practical empirical performance modeling for CFD applications using Extra-P", In Proc. of the 35th International Conference on Parallel Computational Fluid Dynamics (ParCFD2024), Bonn, Germany, pages 1-6, Forschungszentrum Jülich, September 2024.

Last Update: 2025-04-09 11:32