Laboratory for Parallel Programming

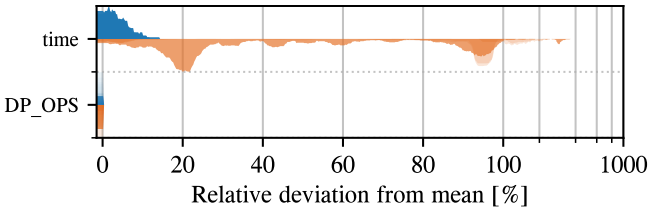
TECHNISCHE UNIVERSITÄT DARMSTADT

Bachelor's / Master's Thesis

Noise Analysis Using Hardware Counters

Motivation

In the dawn of the Exascale systems, scientific applications, as well as the systems they are running on, are increasingly growing in performance and complexity. For a deeper understanding of the application behavior and to identify early performance bottlenecks, massive complex analyses are usually performed, which demand a lot of effort, time, and costs. A much easier and still effective methodology to study the scaling behavior and identify application bottlenecks in an early stage is performance modeling, which has been wildly used in the HPC domain. While performance modeling delivers good insight into the application behavior and its scalability behavior, a lot of factors can affect the models and thus the quality of the analysis. In general, noise can induce performance variability, which is the difference between execution times across repeated runs of an application in the same execution environment. One way to increase the accuracy of the performance models is to utilize hardware counters, as several approaches have shown [1].



Example showing the relative deviation from the arithmetic mean of the floating point operations (DP_OPS) and the runtime on an examined cluster for several benchmarks.

Task

In a recent study, we examined the resilience of hardware counters against noise. Thereby, we classified the counters according to their noise resilience and thus usefulness for boosting performance modeling. Though the study was performed on five large systems, several aspects were missing that will be handled in this thesis. In particular, the output of the thesis shall be an automatic tool that is capable of:

- deploying the setup (benchmarks) and the measurement infrastructure Score-P
- creating and executing the tests with a fallback mechanism
- classifying the hardware counters according to their noise resilience. For example, a machine learning approach performs this classification. The approach must be capable of handling a large amount of data

The portability of the tool is one of the key aspects. The tool should be capable of being deployed on a large cluster with several nodes and a job scheduler (SLURM), as well as on single node systems.
Additional tasks for the master thesis: For the master thesis, an extensive approach for the classification is required. Moreover, several classification methodologies should be developed and compared.

Requirements

- C++ and Python
- Usage of Linux Shell
- Nice to have: Knowledge about Compute Clusters

Contact

Dr. Ahmad Tarraf <ahmad.tarraf@tu-darmstadt.de>

References

[1] Ding, Nan and Lee, Victor W. and Xue, Wei and Zheng, Weimin, APMT:an automatic hardware counter-based performance modeling tool for HPC applications, CCF Transactions on High Performance Computing, Jun 2020

Issue date: **November 2022**