Managing Performance vs. Accuracy Trade-offs with an Efficient Bit-Level precision Tuning

Dorra Ben Khalifa[1] and Matthieu Martel[1,2]

[1] University of Perpignan, LAMPS laboratory, 52 Av. P. Alduy, Perpignan, France
[2] Numalis, Cap Omega, Rond-point Benjamin Franklin, Montpellier, France
{dorra.ben-khalifa,matthieu.martel}@univ-perp.fr

Abstract. Although users of High Performance Computing (HPC) are most interested in raw performance both energy and power consumption has become critical concerns. This is due to several technological issues such as the power limitations of processor technologies and the massive cost of communication which arises while executing applications on such architectures [2]. In recent years, the use of reduced precision to improve the performance metrics is emerging as a new trend to save the resources on the available processors. In practical terms, as almost the numerical computations are performed using floating-point data operations to represent real numbers [1], the precision of the related data types should be adapted in order to guarantee a desired overall rounding error and to strengthen the performance of programs. For instance, using single precision formats (`binary32`) is often at least twice as fast as the double precision (`binary64`) ones. Consequently, the natural question that arises is how to obtain the best precision/performance trade-off by allocating some program variables in low precision (e.g. `binary16` and `binary32`) and by using high precision (e.g.`binary64` and `binary128`) selectively. This process is also called by mixed-precision tuning.
Past research main goal was to improve performance by reducing precision with respect to an accuracy constraint done by static analysis [5,6] or by dynamic analysis [7,8,9]. The major limitation of theses techniques is that they follow a try and fail methodology: they change the data types of some variables of the program and evaluate the accuracy of the result and depending on what is obtained they change more or less data types and repeat the process.
In this article, we present a novel static approach based on a semantical modeling of the propagation of the numerical errors throughout the code. This technique is embodied in an automated tool called `POP`. The main insight of `POP`, in contrast to its former introduction in [2,4,3], is to generate and solve an Integer Linear Problem (ILP) from the program source code. This is done by reasoning on the most significant bit and the number of significant bits of the values which are integer quantities. The optimal solution computed by a classical linear programming solver gives the optimized data types that satisfy the user accuracy requirement in a polynomial time. The originality of `POP`, in comparison with the existing tools, is that our approach find the minimal number of bits needed for each variable, known as bit-level precision tuning to get a certain accuracy on the results. Consequently, our tuning is not dependant

to any particular computer arithmetic (e.g. IEEE754 [1] and POSIT). The main contribution of this article is to demonstrate the effectiveness of `POP` on new benchmarks[3] coming from different application domains. Also, we provide a detailed comparison of `POP` and the state of the art.

Keywords: Static analysis, precision tuning, numerical accuracy, computer arithmetic, integer linear problem.

References

1. ANSI/IEEE: IEEE Standard for Binary Floating-point Arithmetic, std 754-2008 edn. (2008)
2. Ben Khalifa, D., Martel, M.: Precision tuning and internet of things. In: International Conference on Internet of Things, Embedded Systems and Communications, IINTEC 2019. pp. 80–85. IEEE (2019)
3. Ben Khalifa, D., Martel, M.: Precision tuning of an accelerometer-based pedometer algorithm for iot devices. In: International Conference on Internet of Things and Intelligence System, IOTAIS 2020. pp. 113–119. IEEE (2020)
4. Ben Khalifa, D., Martel, M., Adjé, A.: POP: A tuning assistant for mixed-precision floating-point computations. In: Formal Techniques for Safety-Critical Systems - 7th International Workshop, FTSCS 2019. Communications in Computer and Information Science, vol. 1165, pp. 77–94. Springer (2019)
5. Chiang, W., Baranowski, M., Briggs, I., Solovyev, A., Gopalakrishnan, G., Rakamaric, Z.: Rigorous floating-point mixed-precision tuning. In: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL. pp. 300–315. ACM (2017)
6. Darulova, E., Horn, E., Sharma, S.: Sound mixed-precision optimization with rewriting. In: Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS. pp. 208–219. IEEE Computer Society / ACM (2018)
7. Kotipalli, P.V., Singh, R., Wood, P., Laguna, I., Bagchi, S.: AMPT-GA: automatic mixed precision floating point tuning for GPU applications. In: Proceedings of the ACM International Conference on Supercomputing, ICS. pp. 160–170. ACM (2019)
8. Lam, M.O., Hollingsworth, J.K., de Supinski, B.R., LeGendre, M.P.: Automatically adapting programs for mixed-precision floating-point computation. In: International Conference on Supercomputing, ICS'13. pp. 369–378. ACM (2013)
9. Rubio-González, C., Nguyen, C., Nguyen, H.D., Demmel, J., Kahan, W., Sen, K., Bailey, D.H., Iancu, C., Hough, D.: Precimonious: tuning assistant for floating-point precision. In: International Conference for High Performance Computing, Networking, Storage and Analysis, SC'13. pp. 27:1–27:12. ACM (2013)

[3] https://fpbench.org/