Minimal-Precision Computing for High-Performance, Energy-Efficient, and Reliable Computations

Introduction

In numerical computations, the precision of floating-point calculations is a key factor to determine the performance (speed and energy-efficiency) as well as the reliability (accuracy and reproducibility). However, the precision generally plays a contrary role for both. Therefore, the ultimate concept for maximizing both at the same time is the minimal-precision computation through precision-tuning, which adjusts the optimal precision for each operation and data. Several studies have been already conducted for it, but so far, the scope of those studies is limited to the precision-tuning alone. In this study, we propose a more broad concept of the minimal-precision computation with precision-tuning, involving both hardware and software stack.

Our Proposal

Minimal-Precision Computing

Minimal-precision computing is both reliable (aka robust and sustainable) as it ensures the requested accuracy of the result as well as being energy-efficient.

High-performance

Performance can be improved through the minimal-precision as well as fast numerical libraries and accelerators.

Energy-Efficient

Through the minimal-precision as well as energy efficient hardware acceleration with FPGA and GPU.

Reliable

To ensure the requested accuracy, the precision-tuning is processed based on numerical validation, guaranteeing also reproducibility.

General

Our scheme is applicable for any floating-point computations. It contributes to low development cost and sustainability (easy maintenance and system portability).

Comprehensive

We propose a total system from the precision-tuning to the execution of the tuned code, combining heterogeneous hardware and hierarchical software stack.

Realistic

Our system can be realized by combining available in-house technologies.

Our Contributions

A Stochastic Arithmetic Tools

- Discrète Stochastic Arithmetic (DSA) [12] (i) is a technique to model rounding modes (e.g., round up/down/towards nearest, or random), which is applicable by rewriting the code with DSA arithmetic; (ii) is a generic scheme applicable for any floating-point computations, including algorithms and no code modification is needed; it is a lightweight approach in terms of performance, development cost, and development portability. In the present study, it is applied recursively until the output is a random variable. DSA can be implemented as a high-level compiler backend or a library.

- CADNA (Control of Accuracy and Debugging for Numerical Applications) [9] is a DSA library for FORTRAN and C/C++. CADNA can be used on UNIX/Linux/C/C++/Fortran compilers.

- PROMISE (Promising Optimizer) [17] is a DSA-based tool that allows automatically tuning of the precision of arithmetic operations in a program.

- Stochastic Floating-point (SFp) is a DSA library for arbitrary-precision MPFR.

B FPGA as an Arbitrary-Precision Computing Platform

- FPGA enables us to implement arbitrary-precision on hardware, high-level synthesized HLS enables us to program is in HLS, however, compiling arbitrary-precision code and obtaining high-performance is still challenging. High-precision computing with FPGA & GPUs is also a challenge.

- SPGen (Riken) [21]: a compiler to generate low-level hardware code from high-level code. It supports fixed-point and floating-point, it translates high-level arithmetic operations into input code in Standard Instruction Definition (SID). The SIMD uses a data flow graph representation, which is suitable for FPGA. It supports fixed-point only, but it is going to several RHGen to support arbitrary precision. Floating-point is a compiler as a computer supporting arbitrary precision.

- SPGen (Riken) [21]: a compiler to generate low-level hardware code from high-level code. It supports fixed-point and floating-point, it translates high-level arithmetic operations into input code in Standard Instruction Definition (SID). The SIMD uses a data flow graph representation, which is suitable for FPGA. It supports fixed-point only, but it is going to several RHGen to support arbitrary precision. Floating-point is a compiler as a computer supporting arbitrary precision.

- SPGen (Riken) [21]: a compiler to generate low-level hardware code from high-level code. It supports fixed-point and floating-point, it translates high-level arithmetic operations into input code in Standard Instruction Definition (SID). The SIMD uses a data flow graph representation, which is suitable for FPGA. It supports fixed-point only, but it is going to several RHGen to support arbitrary precision. Floating-point is a compiler as a computer supporting arbitrary precision.

- SPGen (Riken) [21]: a compiler to generate low-level hardware code from high-level code. It supports fixed-point and floating-point, it translates high-level arithmetic operations into input code in Standard Instruction Definition (SID). The SIMD uses a data flow graph representation, which is suitable for FPGA. It supports fixed-point only, but it is going to several RHGen to support arbitrary precision. Floating-point is a compiler as a computer supporting arbitrary precision.

C Fast and Accurate Numerical Libraries

Arbitrary-precision arithmetic is used as MPFR or C++ library, but the performance is very low. To overcome this, we employ several other libraries to support arbitrary-precision floating-point operations.

- MPFR [2, 5, 6, 7, 8]: MPFR is a library for multiple-precision floating-point computations with MPFR, which is a reliable, accurate, and portable. (1)

- QPEigen & QPBLAS (ADA, RHGen): QPEigen is a generic eigenvalue solver. QPBLAS (2) is based on double-precision BFPLAS (8). It supports distributed environments with 64-bit environment and takes advantage of shared memory.

- ExBLAS (Seventh University) (11, 12): ExBLAS is an accurate and reliable BLAS built on floating-point expression with error transformations (ITAX, HWV, twa2 and twa1) and error compensation. It supports reproducibility through ensuring correct-statement and preserves error information at the finest level to the detailed format.

- OzBLS ( punishing, RHGen): OzBLS is an accurate and reliable BLAS using Glance scheme (10), which is an accurate arithmetic, multiplication method based on error transformation of product. This scheme is a general and elegant as the usage of the approximations and the outer level (if) and if (const) double version.

Conclusion & Future Work

We proposed a new systematic approach for minimal-precision computations. This approach is reliable, general, comprehensive, high-performance, and realistic. Although the proposed system is still in development, it can be constructed by combining already available developed in-house technologies as well as extending them. Our ongoing step is to demonstrate the system on a small application.

Acknowledgement:

This work is partially supported by the Research Project for Higher Education and Industry 2020 Lead Network Project in Cooperation with Research Project on Advanced Microarchitectures under the Ministry of Education, Culture, Sports, Science and Technology. This work was partly supported by the Japan Science and Technology Agency, Program for the Promotion of Strategic Basic Research Activities, and was supported by the Kemin Foundation. This work was partly supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under Grant-in-Aid for Young Scientists (B) (18700095). The authors gratefully acknowledge all those who contributed to this project.

References:


France-Japan-Germany trilateral workshop: Convergence of HPC and Data Science for Future Extreme Scale Intelligent Applications (Tokyo, Japan) November 6-8, 2019