Offloading Integer GMRES Method to Accelerators

Yijie Yu
yuyijie@g.ecc.u-tokyo.ac.jp
The University of Tokyo
Kashiwa, Chiba, Japan

Toshihiro Hanawa*
hanawa@cc.u-tokyo.ac.jp
The University of Tokyo
Kashiwa, Chiba, Japan

1 INTRODUCTION

Modern computer supports many different precisions, like 64-bit (double) and 32-bit (single). Higher precision is often used as a simple guard against corruption from finite-precision round-off error[1]. However, in many cases, computing tasks can be accomplished by using reduced precision (half or single). Over-allocation of bits resources wastes power, bandwidth, storage, and FLOPS.The transprecision is a solution to the above problem, which uses different precision to execute computing. In addition, as a kind of reconfigurable hardware, FPGA accelerator provides the possibility of higher performance computing which can optimize arithmetic units and data types. We test numerical analysis algorithms like int-GMRES in Host CPU, and offload the kernel to GPU and FPGA in OpenCL and oneAPI to verify the results.

2 INT-GMRES ALGORITHM

Integer arithmetic is useful in the early stage of research and deployment for some computing devices, because floating-point arithmetic are more complex and power consuming. The GMRES method used as a standard solver for sparse unsymmetrical coefficient matrix. And the refinement of int-GMRES is that each FP needs to be converted into integer and fixed point number[2].

Listing 1: Algorithm for int-GMRES

```
Compute r0,v1//FP
For j=1,2,...,m
  Compute wj+1 = A*vj//INT
  For i = 1,...,j
    hi,j = (wj+1,vi)//INT
    wj+1 = wj+1-hi,j*vi//INT
  Endfor
  hi,j = ||wj+1||//INT
  vj+1 = wj+1/hj+1,j//INT
  For i = 1,...,j-1
  Compute Heisenberg matrix
  Endfor
  Compute cj,sj,gj//INT
Endfor
Compute x,b
```

3 ACCELERATOR OFFLOADING

In order to achieve acceleration on hardware, offloading the workload into accelerator is the key. The C/C++ code needs to be divided into host and device code. Host code is executed by the CPU(s) and controls kernel to available devices. Devices correspond to accelerators execute the kernel code. OpenCL is suitable for heterogeneous platforms with strong portability. The DPC++ compiler based on SYCL is a toolkit of oneAPI and supports GPU, CPU, and FPGA. Compared with OpenCL, it supports single source program.

4 EVALUATION

4.1 Host CPU Numerical Experiment

We evaluated the convergence of the relative residual norm of int-GMRES solver in comparison with FP arithmetic in OBCX system. In the early stage of int-GMRES design, long long int type is used for the accuracy. However, it is found that long int also ensures the accuracy. Based on the restart characteristic of int-GMRES, we designed three experiments showed in figure 1, in which the number of iterations is 50, 30 and 20. The third experiment demonstrates the computing time of integer is better than double, and the performance of modified long int is better than long long int type.

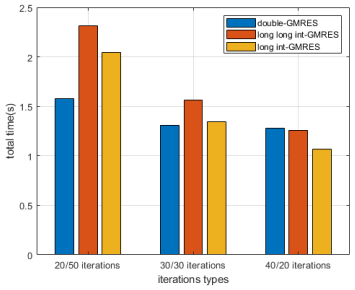

Figure 1: Computation time of three iteration types

4.2 Porting Kernel Code Experiment

Int-GMRES is the kernel code executed in the accelerator. The simplified version of the int-algorithm is shown in Listing 1. The first part of experiment is to execute computing in the GPU in OpenCL for testing, and the other uses DPC++ to offload and compile the kernel on FPGA (Intel Stratix 10). DPC++ compiler supports shift left and right operators which can achieve integer arithmetic.

5 CONCLUSION AND FUTURE WORK

The numerical results demonstrated that lower precision ensures the same convergence performance in int-GMRES.In the future, improving transprecision algorithms to reduce unnecessary computation and developing engine that allow simulations to arbitrary and multiple length precision depending on the tasks are attractive.

REFERENCES

[1] JA Hittinger, PG Lindstrom, H Bhatia, PT Bremer, DM Copeland, KK Chand, AL Fox, GS Lloyd, H Menon, GD Morrison, et al. 2019. *Variable Precision Computing.* Technical Report. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

[2] Takeshi Iwashita, Kengo Suzuki, and Takeshi Fukaya. 2020. An Integer Arithmetic-Based Sparse Linear Solver Using a GMRES Method and Iterative Refinement. In *2020 IEEE/ACM 11th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA).* IEEE, 1–8.