EECE 5640: High Performance Computing

Assignment 3

Jiayun Xin

NUID: 001563582

College of Engineering

Northeastern University Boston, Massachusetts

Fall, 2021

1.

1. The screenshot below shows the computation result of a formular by using Taylor series and indicated that the input with different precision may have different results.

Floats is a 32-bit IEEE 754 single precision (1bit for the sign, 8 bits for the exponent, and 23\* for the value). Float has 7 decimal digits of precision.

Double is a 64-bit IEEE 754 double precision (1 bit for the sign, 11 bits for the exponent, and 52\* bits for the value). Double has 15 decimal digits of precision.

The more decimal digits of precision decide high precision when executing complex computation.

Text

Description automatically generated with medium confidence

1. AVX uses 16 YMM registers and each one can do operations simultaneously on 8 32-bit single-precision floating point numbers or 4 64-bit double-precision floating point numbers [1]. The SIMD register of AVX has 256 bits width. AVX-512 even increases the width of SIMD register into 512 bits. The increased SIMD register width is good for calculating floating-point loop vectorized and dramatically improve task running time.

I run the f(x) function given by q1a for 100000 times, and the running time is listed below. After using “mavx” flag, there are not any different between running time. If I run the Taylor series by using matrix, the process will be speeded up.

Text

Description automatically generated

Text

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | SP | DP |
| 1.1 | 0x3f8ccccd | 0x3FF199999999999A |
| 6200 | 0x45c1c000 | 0x40B8380000000000 |
| -0.044 | 0xbd343958 | 0xBFA6872B020C49BA |

2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| THREADS\_NUM | 1 | 2 | 4 | 8 | 16 |
| Running time(s) | 0.11 | 0.12 | 0.15 | 0.2 | 0.31 |

The table shows the program running time corresponding to the threads number. When I move from 8 to 16 threads, the running time increases. It may be affected by the switch with different threads.

3.

a) Compared with pthread, using OpenMP is easier to do multithread work because of its high level and less code needed. The whole code style is more tidy and easier to read.

b)

Text

Description automatically generated with medium confidence

Text

Description automatically generated

The top one shows the result of the Dining Philosophers problem using pthread and the bottom is the result using OpenMP. Comparing both results, philosophers have different eating order because of the difference of threads about running time. In addition, I used M1 operation system to execute pthread DP problem and Discovery to do OpenMP DP problem, which can be another reason causing the difference of result.

4.

Text

Description automatically generated with medium confidence

Text

Description automatically generated with low confidence

As screenshot shows, I use 8 threads to execute the vector-matrix multiplication.

A close-up of a document

Description automatically generated with low confidence

The screenshot shows the output of vector-matrix multiplication. I set all number in matrix as 1 and vector elements from 1 to 511.

5.

Unit Vector Block is a novel sparse matrix format introduced by Kebing Wang, Bianny Bian and Yan Hao in 2019 [2]. Compared with CSR or CSC format, it needs less memory capacity because this sparse matrix storage method does not store column or row index. Moreover, it improves the SpMV algorithm performance by using a single bitwise manipulation instruction and make full use of Intel AVX-512 instruction set [2].

References:

[1] sourced from: <https://en.wikipedia.org/wiki/Advanced_Vector_Extensions>

[2] K. Wang, B. Bian and Y. Hao, "Innovative Unit-Vector-Block Storage Format of Sparse Matrix and Vector," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), 2019, pp. 351-354, doi: 10.1109/CCOMS.2019.8821708.