Scientific Computing and Machine Learning on Multi- and Manycore Architectures Exercise 2

林澤佑
Tse-Yu Lin

D06946003@ntu.edu.tw

Data Science Degree Program

- ## Analysis

The different between GFLOPS/s performace of LU decomposition without and with pivoting are shown below.

First, both roofline are significant when the size of matrix over 300.

Seond, he enhanced LU decomposition is based on the column pivoting. Besides, instead of swapping columns for pivoting, I use an column index array to record the order of column swaping, That is, I record the permuatation matrix P in PLU decomposition in a compact form.

With the reason above, the calculation of matrix updating relies on the substitution of index array. This causes the performance of LU decomposition with pivoting has a constant decay compared with the one without pivoting. In this case, the performance with pivoting is around 0.7 of the one without pivoting.

Note that in this exercise, the matrix A generated by the program is Poisson-type matrix. That is, A is of the form

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & & \cdots & & 1 & -2 \end{bmatrix}$$

For step i in LU decomposition, pivot column is always in column $i+1$. Hence, the index array at the end is of the form:

$$\begin{bmatrix} 1 & 2 & 3 & \cdots & N-1 & 0 \end{bmatrix}$$

Run the code

For each program in this exercise, please use

```
gcc ex2_x.c -o ex2_x -lm
```

to compile, where $x = 1, 3$ indicates the LU decomposition without and with pivoting.

To see computational time, please type

```
./ex2_x N
```

where N is the size of matrix/vector used for computation. The result would looks like:

```
Time : 0.003655 sec
```

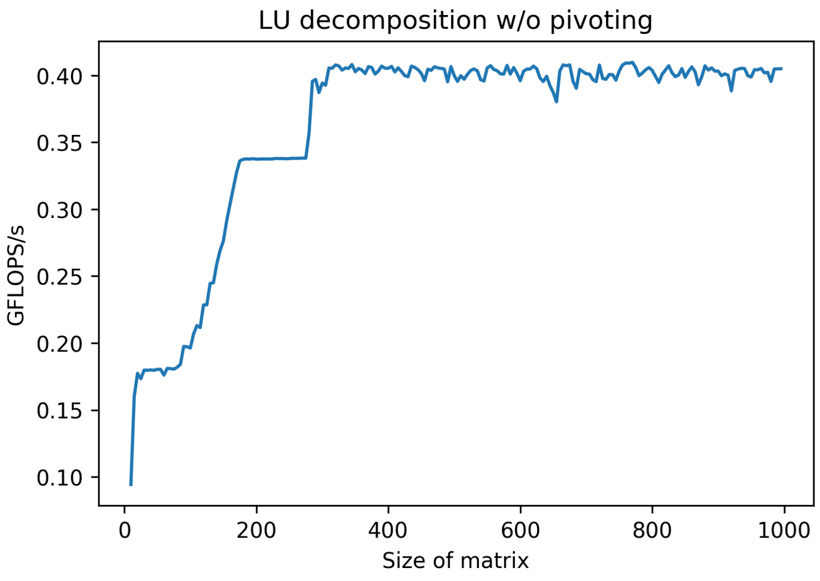

Figure 1: Performace of LU decomposition without pivoting.

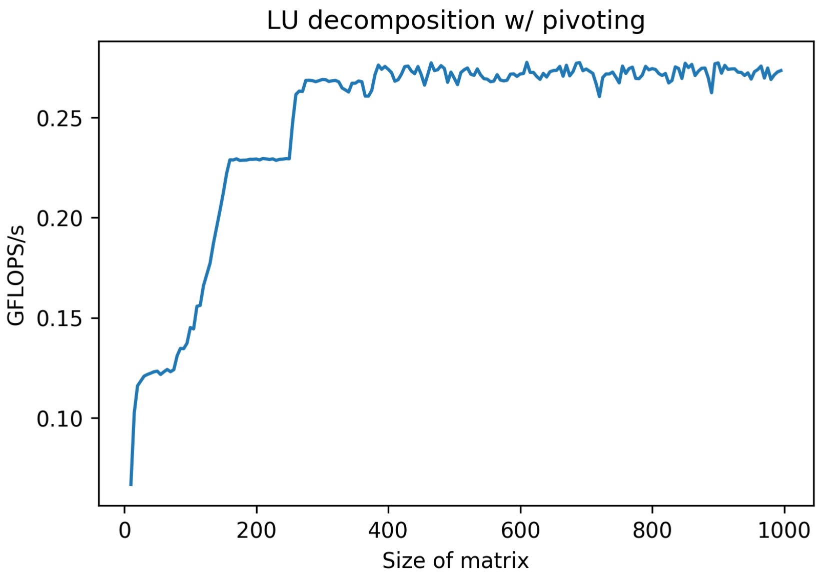

Figure 2: Performace of LU decomposition with pivoting.