LU Decomposition

By
Dilip Puri
Govind Meena
Hemant Kumar

November 15, 2016

Problem

We have a matrix A, now we want to decompose matrix A into two matrices L(Lower Triangular Matrix) and U(Upper Triangular Matrix) such that

$$\mathbf{A = L * U}$$

where

$$L = \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix}, U = \begin{bmatrix} d & e & f \\ 0 & g & h \\ 0 & 0 & i \end{bmatrix}.$$

Motivation

So the first question come to mind that why we are doing this?
so straight forward answer would be that it will simplify things.
How?
Most of the time in mathematics modeling we came up with system of linear equations in the form of

$$\mathbf{Ax = b}$$

so finding A^{-1} is quite difficult so we will use LU decomposition

How?

Example

Lets we have system of eqations

$$[A]\{x\} = \{b\}$$
$$[L][U]\{x\} = \{b\}$$
$$(\therefore [A] = [L][U])$$
$$\{y\} = [U]\{x\}$$
$$[L]\{y\} = \{b\}$$
$$\because [] = matrix, \{\} = vector$$

This will make our system so simple to solve...

Example

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 13 \\ 4 \end{bmatrix}.$$

$$A = LU \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix} * \begin{bmatrix} d & e & f \\ 0 & g & h \\ 0 & 0 & i \end{bmatrix} = \begin{bmatrix} d & e & f \\ ad & ae+g & af+h \\ bd & be+cg & bf+ch+i \end{bmatrix}.$$

$$\Rightarrow \begin{bmatrix} d & e & f \\ ad & ae+g & af+h \\ bd & be+cg & bf+ch+i \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}$$

Now compare the values and get the values of elements of L and U.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}, U = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}.$$

The next step is to solve [L]{y}={b} for the vector {y} that we consider

$$Ly = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 13 \\ 4 \end{bmatrix} = b$$

which can be solved by forward substitution $\{y\} = [3\ 4\ -6]^T$ now that we have found y we finish the procedure by solving

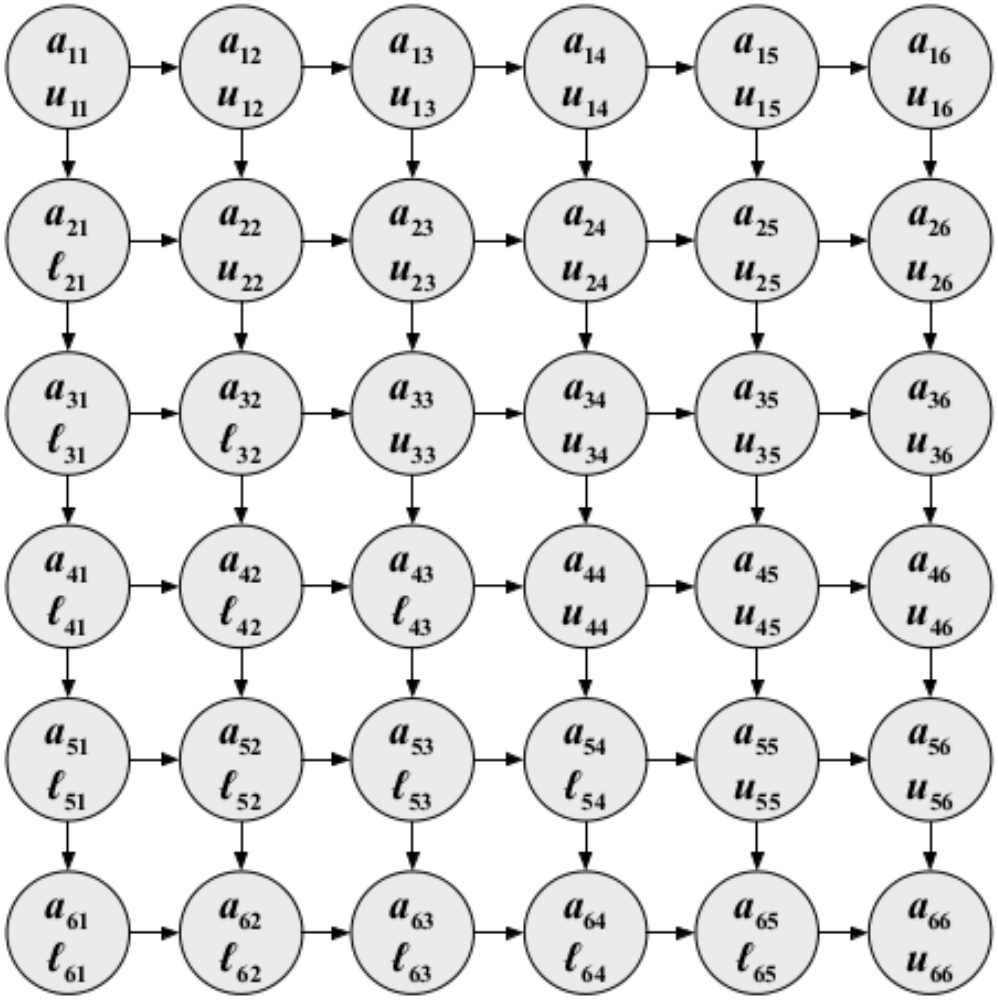
$$Ux = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ -6 \end{bmatrix} = y$$

by using backward substitution we will get {x}.

Algorithm

for $k = 1$ **to** $\min(i, j) - 1$
  recv broadcast of a_{kj} from task (k, j) { vert bcast }
  recv broadcast of ℓ_{ik} from task (i, k) { horiz bcast }
  $a_{ij} = a_{ij} - \ell_{ik}\, a_{kj}$ { update entry
end
if $i \leq j$ **then**
  broadcast a_{ij} to tasks (k, j), $k = i + 1, \ldots, n$ { vert bcast }
else
  recv broadcast of a_{jj} from task (j, j) { vert bcast }
  $\ell_{ij} = a_{ij}/a_{jj}$ { multiplier }
  broadcast ℓ_{ij} to tasks (i, k), $k = j + 1, \ldots, n$ { horiz bcast }
end

Task Generation and Dependency Graph



Since the LU decomposition is completely done using elimination and substation, there are 3 for loops involved and also the cells of a particular row (for U) or particular column (for L) can be computed in parallel so we used LU decomposition block as a suitable area to implement parallelism. Each code has sufficient comments inside to describe the methods and important statements.

SpeedUp Table

Matrix Size\#of Threads	3	10	100	1000
2	1	1	0.9600804136	0.999864061
4	2	1.2	2.0184150943	1.0136727589
8	4	1.3846153846	2.0932999374	1.0098715415
16	4	1.2857142857	2.1128140307	1.0087423196
32	4	1.2857142857	2.1128140307	1.0095692982

Figure : SpeedUP Table

SpeedUp Graph

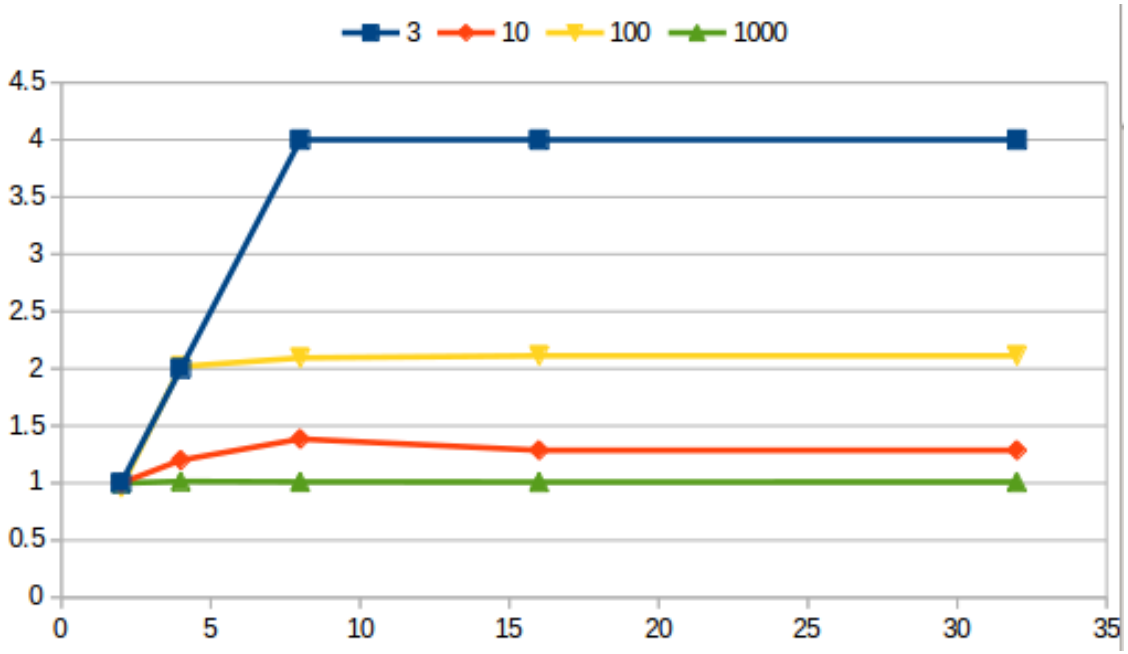


Figure : SpeedUp Graph

Performance

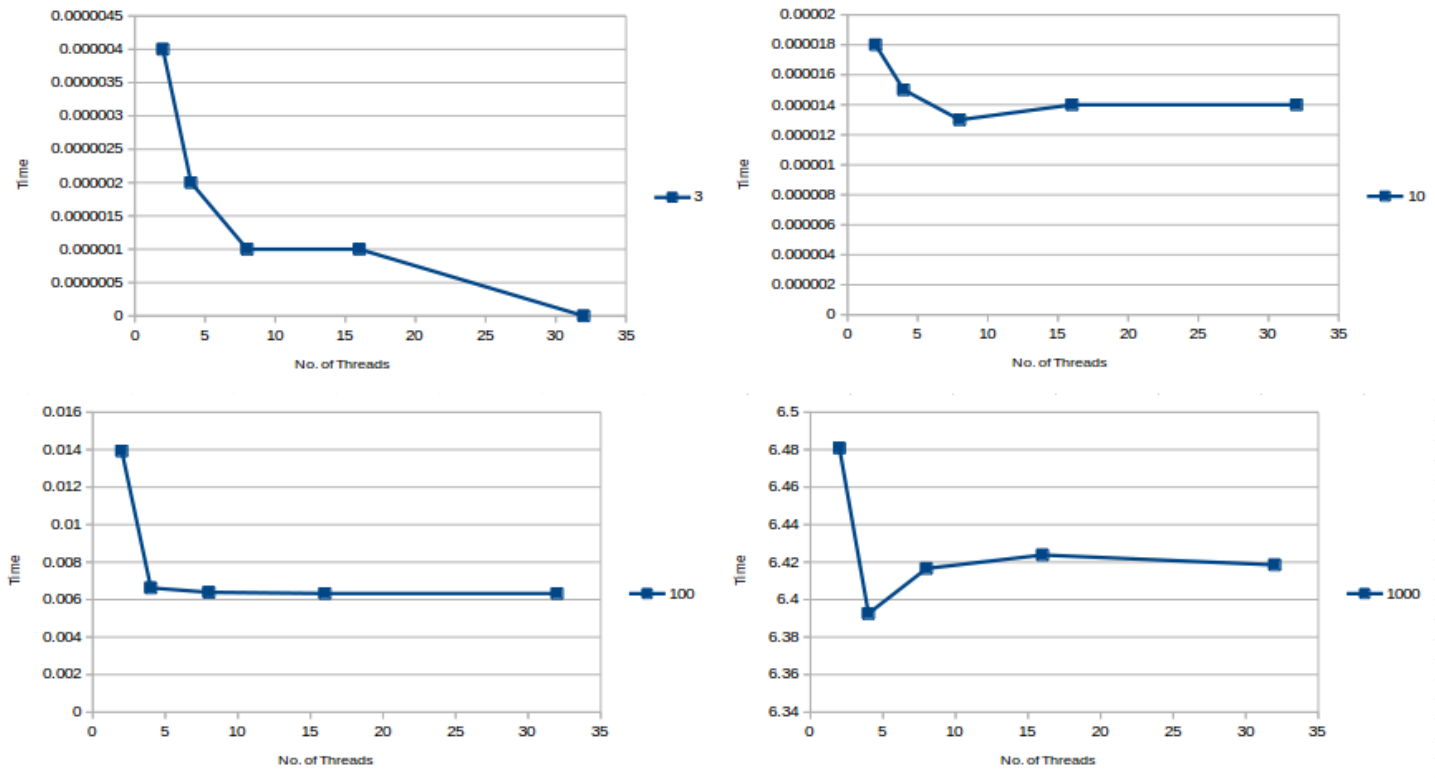


Figure : No. of Threads vs. Time

CPU Usage

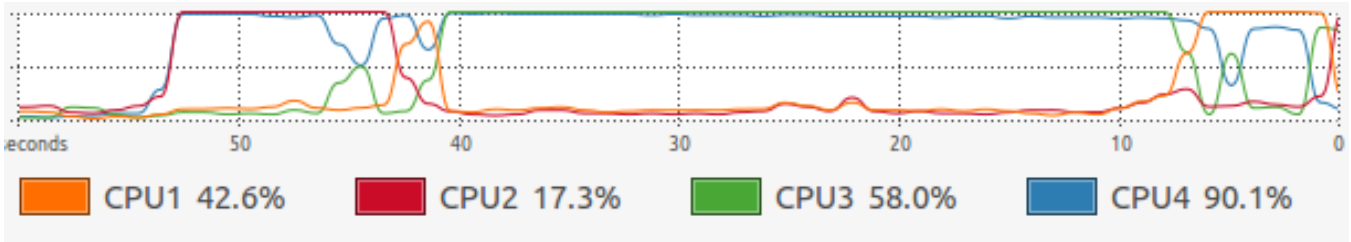


Figure : For 2 threads

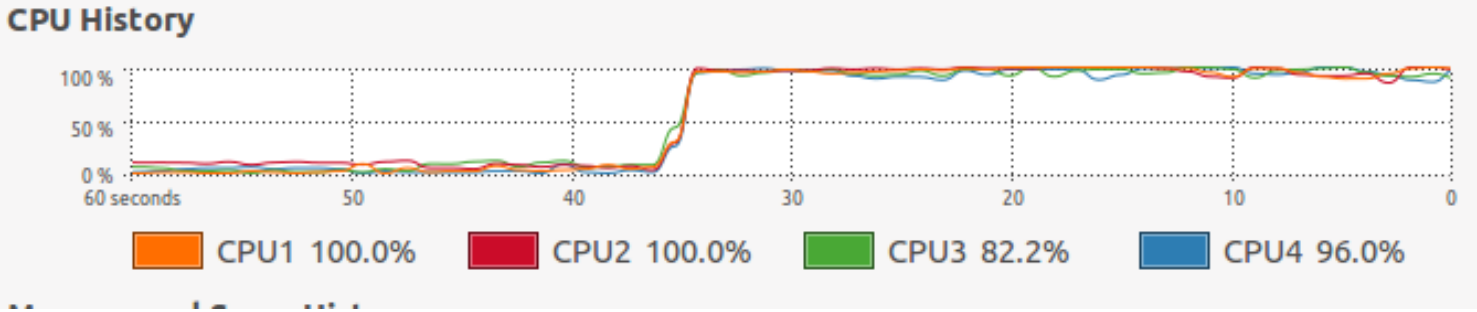


Figure : For 4 threads

Call Map

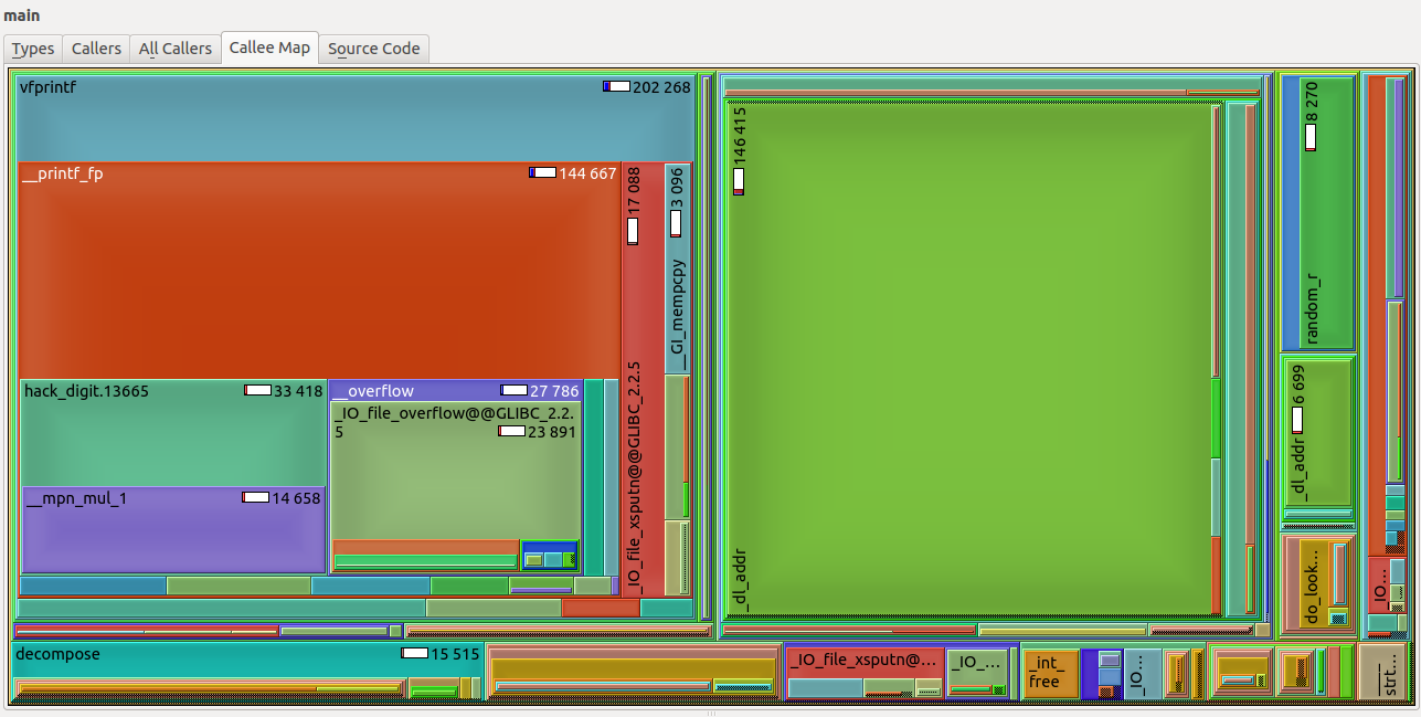


Figure : Callee Map for Parallel code

Cache Misses

```
==5581==
==5581== Events    : Ir Dr Dw I1mr D1mr D1mw ILmr DLmr DLmw
==5581== Collected : 501008 125852 65614 1323 3200 778 1297 2259 687
==5581==
==5581== I   refs:      501,008
==5581== I1  misses:      1,323
==5581== LLi misses:      1,297
==5581== I1  miss rate:    0.26%
==5581== LLi miss rate:    0.25%
==5581==
==5581== D   refs:      191,466  (125,852 rd + 65,614 wr)
==5581== D1  misses:      3,978  (  3,200 rd +    778 wr)
==5581== LLd misses:      2,946  (  2,259 rd +    687 wr)
==5581== D1  miss rate:     2.0% (    2.5%   +    1.1%  )
==5581== LLd miss rate:     1.5% (    1.7%   +    1.0%  )
==5581==
==5581== LL refs:         5,301  (  4,523 rd +    778 wr)
==5581== LL misses:       4,243  (  3,556 rd +    687 wr)
==5581== LL miss rate:      0.6% (    0.5%   +    1.0%  )
```

Figure : Cache Misses

Q & A

Thank You!