Look ahead and Depths

Official description:
Once the panel has been broadcast or say during this broadcast operation, the trailing sub matrix is updated using the last panel in the look-ahead pipe: as mentioned before, the panel factorization lies on the critical path, which means that when the kth panel has been factorized and then broadcast, the next most urgent task to complete is the factorization and broadcast of the k+1 the panel. This technique is often referred to as "look-ahead" or "send-ahead" in the literature. This package allows to select various "depth" of look-ahead. By convention, a depth of zero corresponds to no lookahead, in which case the trailing sub matrix is updated by the panel currently broadcast. Look-ahead consumes some extra memory to essentially keep all the panels of columns currently in the look-ahead pipe. A look-ahead of depth 1 (maybe 2) is likely to achieve the best performance gain.

what is a “Panel”

- During the benchmark while the process of LU decomposition is happening, a panel is a block or subset of columns of the matrix that is being factorized. This matrix is divided into multiple panels usually and each panel is processed one at a time during the LU Decomposition
- The matrix is divided into panels to use parallel computation. Each panels is referred both size of the blocks controlled by the parameter NBs
- EX// if the matrix Is N X N and NBs = 4, then each panel will contain 4 columns of the matrix
- It is factorized panel by panel.
- After the first matrix is factorized, that matrix is used to update the trailing sub matrix(the one forward of it that has not been factorized yet). Then the trailing sub-matrix is updated after all the LU factors of the panel has been broadcast to all processors

Terms
factorize: The process of decomposing a matrix into its component factors, typically for solving linear systems or eigenvalue problems.
Broadcast: The distribution of data (usually a matrix or vector) from one processor to others in a parallel computing environment.
Trailing sub matrix: The portion of a matrix that remains after removing the current panel, typically used for updating the matrix during factorization.

Example of panel
A =

```
    | a11 a12 a13 a14 a15 a16 |
    | a21 a22 a23 a24 a25 a26 |
```

| a31 a32 a33 a34 a35 a36 |
| a41 a42 a43 a44 a45 a46 |
| a51 a52 a53 a54 a55 a56 |
| a61 a62 a63 a64 a65 a66 |

If the block size (NBs) is 3 you might divide the matrix into two panels:
Panel 1 =

| a11 a12 a13 |
| a21 a22 a23 |
| a31 a32 a33 |

Panel 2 =

| a14 a15 a16 |
| a24 a25 a26 |
| a34 a35 a36 |

Then 2 more panels and so on

The process of look ahead (no look ahead, look ahead Depth=1, look ahead Depth=2

No Look Ahead Process

Time Step	Processor 1	Processor 2	Processor 3	Processor 4
T1	Factorize Panel 1			
T2	Broadcast Panel 1			
T3	Update trailing submatrix with Panel 1			
T4	Factorize Panel 2			
T5	Broadcast Panel 2			
T6	Update trailing submatrix with Panel 2			

Look Ahead Depth = 1

Time Step	Processor 1	Processor 2	Processor 3	Processor 4
T1	Factorize Panel 1			
T2	Broadcast Panel 1	Factorize Panel 2		
T3	Update trailing submatrix with Panel 1	Broadcast Panel 2	Factorize Panel 3	
T4	Update trailing submatrix with Panel 2			

Look Ahead Depth = 2

Time Step	Processor 1	Processor 2	Processor 3	Processor 4
T1	Factorize Panel 1			
T2	Broadcast Panel 1	Factorize Panel 2	Factorize Panel 3	
T3	Update trailing submatrix with Panel 1	Broadcast Panel 2	Factorize Panel 4	Factorize Panel 5
T4	Update trailing submatrix with Panel 2	Broadcast Panel 3	Update trailing submatrix with Panel 3	Broadcast Panel 4
T5	Update trailing submatrix with Panel 4			

Back to the example:
We have a 6x6

No look ahead

No look ahead will factorize the first panel then after factorization will broadcast to all processors in the process. Then update the trailing sub matrix which is the next panel. This is all one process and creates idle time so more time can be used to factor the panels while others are finishing up

Depth=1
Depth 1 starts with Factorizing panel 1. Then if panel 1 is factorized then panel 2 will start factorizing. By then panel 1 will also start broadcasting at the same time. Then panel 1 will be updating, panel 2 will be broadcasting and panel 3 will start factorizing.

Depth2 =
Depth 2 starting with factoring panel 1 then does the same as depth = 1 but instead will look ahead into 2 panels and start their factorization

Which one is more efficient:
No look ahead:

- Simple, minimizes data dependencies
- Least efficient in parallelism. No overlap. Allows for idle time that can be used elsewhere in the process

Depth=1:

- Offers some parallelism by overlapping certain operations, better than no look ahead, improves performance.
- It turns out that a depth of 1 seems to give the best results, but may need a large problem size before one can see the performance gain

Depth=2

- A look-ahead of depth 1 (maybe 2) is likely to achieve the best performance gain.
- Maximizes parallelism by allowing multiple operations to overlap
- More complex, higher communication overhead between processors

Depth = 3 and more

- Increase communication overhead
- Higher sync frequency across all processors
- At a certain point having more depth does not improve results
- Recommended to keep it at 1 and 2 for best results