Dense LU Factorization

Dr.N.Sairam & Dr.R.Seethalakshmi
School of Computing,
SASTRA Univeristy,
Thanjavur-613401.

Contents

1. Dense LU Factorization

Dense LU Factorization is a popular benchmark to evaluate the performance of parallel(super) computers. It is an important parallel kernel due to its use in scientific applications. Its performance is primarily computation bound. Its performance is excellent with respect to memory allocation and access, communication and scheduling. These lead to higher efficient systems.

Dense LU Factorization is used a mechanism to solve dense linear systems. A set of n linear equations in n variables is solved by performing LU Factorization and solving the triangular system which is obtained. The algorithms have different variations. To mention, we have Crout's algorithm which performs an in-place computation. Numerical stability is derived with partial pivoting. Most formulations of LU are blocked algorithms with underlying sequential operations delegated to a high performance linear algebra library.

In computer field major implementations use block-cyclic distributions of matrix blocks onto a two-dimensional process grid. The process grid dimensions derive a trade-off between communication and computation and are architecture- and implementation-sensitive. The critical panel factorization steps can be made less communication-bound by overlapping asynchronous collectives for pivoting with the computation of rank-k updates. By shifting the computation communication trade-off, a modified block-cyclic distribution can beneficially exploit more available parallelism on the critical path and reduce panel factorization's memory hierarchy contention on now-ubiquitous multicore architectures.

In any matrix dealing we consider row-major and column-major composition. In the column-major process grid, the performance is reduced as too many processors contend for the memory access. In the row-major process grid, the performance is not degraded, but it sacrifices node and network locality in the critical pivoting steps. To overcome these two problems, third method called striding is used. Maximum available parallelism

is achieved by the length or height of the process grid. To start with, a taller grid is considered and later it is rotated to obtain the advantages of row major form.

Let us assume an nXn matrix which is decomposed into block size of b^2 which is distributed across the grid.

1.2 Algorithm

Dense LU Factorization Algorithm

The factorization process can be described as follows:
For step 0 to n-1
A) Active panel blocks are those at/below diagonal block step
1) for column in 0::b: (on each active panel block)

a) Each block identifies its maximum value below the diagonal in the current column within that block and contributes to a reduction among the active panel blocks.

b) The resut of the reduction identifies the pivot row, which is swapped to the diagonal

position and broadcast to all of the active panel blocks.

c) Each active panel block performs a rank-1 update of the section after column with multipliers from column and the pivot row.

2) The sequence of pivot exchanges is broadcast to the blocks of U and the trailing submatrix,
Which communicate to apply the same swaps as the active panel.
3) Active panel blocks send their contents, each a portion of L, to the blocks to their right.
4) U blocks to the right of the diagonal each perform a triangular solve, and send the result to the blocks below them.
5) Blocks in the trailing sub matrix each compute a trailing update as the product of the L and U blocks they have received.

B) Data Distribution

There are several aspects that are to be considered when distributing matrix blocks onto the available processors. The amount of computation that has to be performed on each block of data to achieve the final factorization differs based on the location of the block in the overall matrix. The accumulated wisdom in the community has shown that a two dimensional block-cyclic data distribution achieves satisfactory
load balance while also presenting good communication behavior and stability.

For such a data distribution, the available processes (N), are first arranged into a two-dimensional process grid of dimensions PXQ such that N = P XQ. The aspect ratio of the process grid and how the process ranks are arranged within this grid depend on system architecture. The two dimensional block cyclic pattern can be obtained by tiling the process grid onto the matrix as shown in Figure 1(a).
The figure depicts a matrix decomposed into blocks. The bold lines are the boundaries of the process grid and the label in each block is the process rank to which it is mapped. Expressions for the traditional block-cyclic mapping of a block, whose coordinates in the blocked view of the input matrix is (x; y), onto a process grid of dimensions PXQ are as follows2:
m1 = x mod P (x in grid)
n1 = y mod Q (y in grid)
frow(x; y; P;Q) = m1Q + n1 ----------(1)
fcol(x; y; P;Q) = n1P + m1 ----------(2)
Equation 1 yields the process rank for a row-major process grid, while equation 2 yields the process rank for a column-major process grid.
c) Granularity Spectrum
The factorization presents a challenging spectrum of computation and communication grain sizes. The trailing updates comprise the bulk of the computation in a dense LU solver.Each trailing update is an O(b3) matrix-matrix multiplication(i.e. a call to the dgemm() level-3 BLAS routine). The triangular solves (via dtrsm()) are of similar

computational cost. Each trailing update or triangular solve takes tens of milliseconds for the block sizes common on today's architectures. Large messages drive the heavy computation kernels.In contrast, the active panel is communication intensive, with b small-message pivot reductions and broadcasts occurring in rapid succession, interspersed with smaller computations. Eachof the se fine-grained steps on the active panel nominally takehundreds of microseconds to single digit milliseconds.

D)Lookahead

Ideally, every processor would remain busy during the entirefactorization process. However, in each step, only a subset of processors own blocks that participate in the active panel.Thus, to avoid idling processors, work from multiple stepsmust be overlapped. The extent of the overlap (specifically,the number of steps that the active panel runs ahead of trailingupdates) in an implementation of dense LU is known as itslookahead depth.