Indian Institute of Information Technology Allahabad

Project-1 - PP (Nov 2021)

M.Tech

Course Name	Course Code	Deadline
Programming Practices	—-	—–

Important Instructions: *This project can be done in a team of two students. Teams will be created by the students itself and then has to be informed to the TA within two days of announcement of the project. Thanks to Dr. Rijurekha Sen IIT Delhi for the original assignment.*

In this project, you will develop two parallel implementations (using PThreads and OpenMP) of LU decomposition that use Gaussian elimination to factor a dense $N \times N$ matrix into an upper-triangular one and a lower-triangular one. In matrix computations, pivoting involves finding the largest magnitude value in a row, column, or both and then interchanging rows and/or columns in the matrix for the next step in the algorithm. The purpose of pivoting is to reduce round-off error, which enhances numerical stability. In your assignment, you will use row pivoting, a form of pivoting involves interchanging rows of a trailing submatrix based on the largest value in the current column. To perform LU decomposition with row pivoting, you will compute a permutation matrix P such that `PA = LU`. The permutation matrix keeps track of row exchanges performed.

Below is pseudocode for a sequential implementation of LU decomposition with row pivoting.

```
inputs: a(n,n)
outputs: p(n), l(n,n), and u(n,n)

initialize p as a vector of length n
initialize u as an n x n matrix with 0s below the diagonal
initialize l as an n x n matrix with 1s on the diagonal and 0s above the diagonal
for i = 1 to n
   p[i] = i
for k = 1 to n
   max = 0
   for i = k to n
      if max < |a(i,k)|
         max = |a(i,k)|
         k' = i
   if max == 0
      error (singular matrix)
   swap p[k] and p[k']
   swap a(k,:) and a(k',:)
   swap l(k,1:k-1) and l(k',1:k-1)
   u(k,k) = a(k,k)
   for i = k+1 to n
      l(i,k) = a(i,k)/u(k,k)
      u(k,i) = a(k,i)
   for i = k+1 to n
      for j = k+1 to n
         a(i,j) = a(i,j) - l(i,k)*u(k,j)
```

```
Here, the vector p is a compact representation of a permutation matrix p(n,n),
which is very sparse. For the ith row of p, p(i) stores the column index of
the sole position that contains a 1.
```

You will write two shared-memory parallel programs that perform LU decomposition using row pivoting. You will develop one solution using the Pthreads programming model and one using OpenMP.

Each LU decomposition implementation should accept two arguments: n: the size of a matrix, followed by t: the number of threads. Your programs will allocate an $n \times n$ matrix a of double precision (64-bit) floating point variables. You should initialize the matrix with uniform random numbers computed using a suitable random number generator, such as drand48, drand48_r, or the C++11 facilities for pseudo-random number generation. (Note: if you are generating random numbers in parallel, you will need to use a reentrant random number generator and seed the random number generator for each thread differently.) Apply LU decomposition with partial pivoting to factor the matrix into an upper-triangular one and a lower-triangular one.

To check your answer, compute the sum of Euclidean norms of the columns of the residual matrix (this sum is known as the L2,1 norm) computed as PA-LU. Print the value of the L2,1 norm of the residual. (It should be very small.)

The verification step need not be parallelized. Have your program time the LU decomposition phase by reading the real-time clock before and after and printing the difference.