MPI with pthread implementation of Gaussian Elimination

Gabriel Dimitriu

1 Introduction

Let the system be $Ax = b$. Here A is a dense n x n matrix of coefficients such that $A[i][j] = a_{i,j}$, b is an n x 1 vector and x is the desired solution vector.

A system of equations is solved in two stages. First, through a series of algebraic manipulations, the original system of equations is reduced to an upper triangular system of the form $Ux = y$, where U is a unit upper-triangular matrix. In the second stage of solving a system of linear equations, the upper-triangular system is solved for the variables in reverse order from x_{n-1} to x_0 by a procedure know as back-substitution.

Algorithm 1 *gaussian_ elimination*

```
1.  procedure gaussian_elimination(A,b,y)
2.  begin
3.      for k:=0 to n-1 do
4.              for j:=k+1 to n-1 do
5.                  A[k][j]:=A[k][j]/A[k][k]; //division step
6.          y[k]:=b[k]/A[k][k];
7.          A[k][k]:=1;
8.          for i:=k+1 to n-1 do
9.                  for j:=k+1 to n-1 do
10.                 A[i][j]:=A[i][j]-A[i]k]xA[k][j]; //elimination step
11.             b[i]:=b[i]-A[i][k]xy[k];
12.             A[i][k]:=0;
13.         endfor;
14.     endfor;
15. end gaussian_elimination;
```

2 Pipeline communication and Computation

The matrix is scattered to all processor so two consecutive row are in two consecutive processors.

During the k^{th} iteration processor P_k broadcast part of the k^{th} row of the matrix to processors $P_{k+1}, ..., P_{p-1}$. Assume that the processors $P_0 ... P_{p-1}$ are connected in a linear array, and P_{k+1} is the first processor to receive the k^{th} row from processor P_k. Then the processor P_{k+1} must forward this data to P_{k+2}. However, after forwarding the k^{th} row to P_{k+2}, processor P_{k+1} need not wait to perform the elimination step until all the processors up to P_{p-1} have received the k^{th} row. Similarly, P_{k+2} can start its computation as soon as is has forwarded the k^{th} row to P_{k+3}, and so on. Meanwhile, after completing the computation for the k^{th} iteration, P_{k+1} can perform the division step, and start the broadcast of the $(k+1)^{th}$ row by sending it to P_{k+2}. The forwarding communication is done in a separate posix thread to eliminate the overhead and in the same time to not use a not-blocked communication. The synchronization between receiving and elimination step is done using semaphore because the thread is alive until the processor enter in the division step.

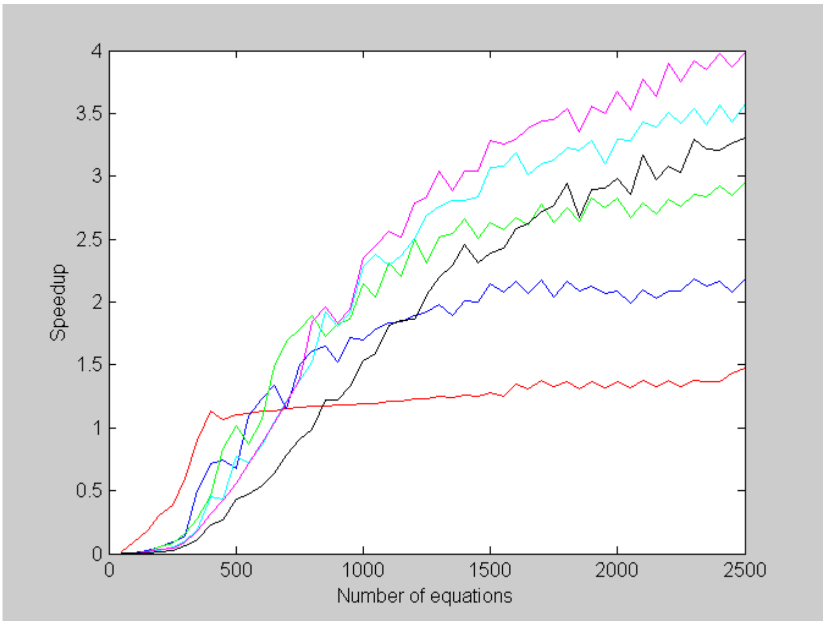
3 Results

The benchmark is made on a 100MBs network with Fedora Core 1 and realteck 8139 network card. The MPI version is MPICH 1.2.6 with native gcc and rsh authorization.

The network consists in 1 PII-500MHz, 2 PIII-800MHz, 1 athlon-1200MHz, 1 PIV-1400MHz and 1 dual PII-500MHz and the speedup is the average of ten runs.

The Gaussian Elimination is implemented without load balancing so because the fist and in the same time the slowest computer is PII-500MHz so the reference of the speedup is this computer.

In the following graph is presented the speedup with red for two computers, blue for three computers, green for four computers, cyan for five computers, magenta for six computers and black for seven computers.


Speedup
Number of equations