Assignment #06

Nov, 20, 2017

JinYeong Wang
Parallel Computing Lab.
Mechanical Engineering
Hanyang University

Problem Definition

- Compute

$$\mathbf{Ax} = \mathbf{y}$$

- While

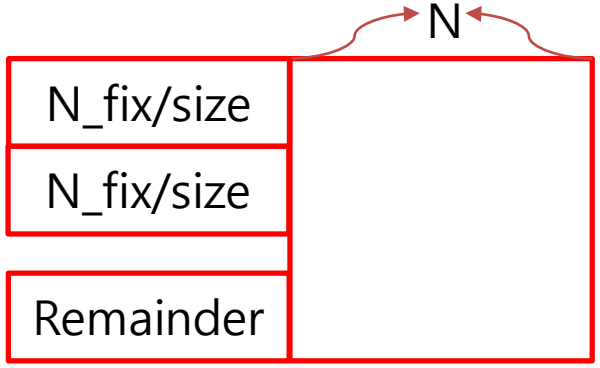
$$\mathbf{A} = N\ by\ N\ Matrix$$

$$\mathbf{x}\ and\ \mathbf{y} = N\ by\ 1\ Vector$$

- Using
  - MPI_Gatherv

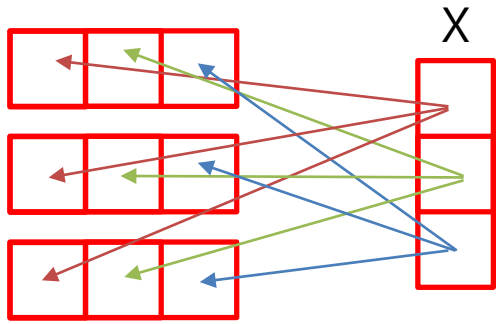
Methodology

Matrix Divide



- If N % size != 0
  - Make N_fix
  - Increase N_fix till N_fix%size ==0

Gatherv



- Each Processor has N by 1 vector for using MPI_Gatherv for gathering X vector

Example when N =3 When 2 Processors

$$\begin{bmatrix} 4 & 5 & 5 \\ 2 & 7 & 9 \\ 7 & 2 & 7 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \\ 3 \end{bmatrix} = [Unknown]$$

- In rank 0
  - A = $\begin{bmatrix} 4 & 5 & 5 \\ 2 & 7 & 9 \end{bmatrix}$
  - X = $\begin{bmatrix} 7 \\ 4 \end{bmatrix}$
- In rank 1
  - A = $[7 \quad 2 \quad 7]$
  - X = $[3]$
- Gather X to Temp
  - Each Processors will has Entire X
  - Temp = $\begin{bmatrix} 7 \\ 4 \\ 3 \end{bmatrix}$

Code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <mpi.h>
#include <math.h>
int main(int argc, char **argv)
{
    int rank, size;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    int i, j, N, N_fix, cols_p, rows_p, sum = 0;
    double tic, toc;
    if(rank == 0)
    {
        fflush(stdin);
        printf("It will compute A * x = y\n");
        printf("A will be N by N Matrix, x and y will be N by 1 Matrix.\n");
        printf("Input N\n");
        scanf("%d", &N);
        N_fix = N;
        while(N_fix%size!=0)
            N_fix++;
    }
    MPI_Bcast(&N,1,MPI_INT,0,MPI_COMM_WORLD);
    MPI_Bcast(&N_fix,1,MPI_INT,0,MPI_COMM_WORLD);
    cols_p = N;
    rows_p = N_fix/size;
    if(rank == size-1)
        rows_p = N-rows_p*(size-1);
```

Compute cols_p ans rows_p for Memory allocation

Code

Memory allocation for A, x, y and temp for gather x

```
int *A_p = (int *)calloc(rows_p * cols_p, sizeof(int));
int *x_p = (int *)calloc(rows_p * 1, sizeof(int));
int *temp_p = (int *)calloc(1 * cols_p, sizeof(int));
int *y_p = (int *)calloc(rows_p * 1, sizeof(int));
int *recvcounts = malloc(sizeof(int)*size);
int *displs = malloc(sizeof(int)*size);
// Define Matrix
srand(time(NULL));
for( j = 0 ; j < rows_p ; j++)
{
    for( i = 0 ; i < cols_p ; i++)
    {
        int coord = j * cols_p + i;
        A_p[coord] = (rand()/(rank+1));
    }
    x_p[j] = (rand()/(rank+1));
}
for( i = 0 ; i < size-1 ; i++ )
{
    recvcounts[i] = N_fix/size;
    displs[i] = sum;
    sum += recvcounts[i];
}
recvcounts[size-1] = cols_p-sum;
displs[size-1] = sum;
```

Memory allocation for Using MPI_Gatherv

Input Random Number to A and x

Input Parameter for MPI_Gatherv

Code

```
    int rank_check;
    MPI_Barrier(MPI_COMM_WORLD);
    if(rank == 0)
        tic = MPI_Wtime();
    for(rank_check = 0 ; rank_check < size ; rank_check++)
    {
        MPI_Gatherv(x_p, recvcounts[rank], MPI_INT, temp_p, recvcounts, displs, MPI_INT, rank_check, MPI_COMM_WORLD);
    }
    for( j = 0 ; j < rows_p ; j++ )
    {
        for( i = 0 ; i < cols_p ; i++)
        {
            int coord = j * cols_p + i;
            y_p[j] += A_p[coord] * temp_p[i];
        }
    }
    MPI_Barrier(MPI_COMM_WORLD);
    if(rank == 0)
    {
        toc = MPI_Wtime();
        printf("Total Excute Time = %lf seconds\n", toc-tic);
    }
    printf("\n");
    free(A_p);
    free(x_p);
    free(y_p);
    free(recvcounts);
    free(temp_p);
    free(displs);
    MPI_Finalize();
}
```

Main Computation
MPI_Gatherv and Compute y

Validate Result

Code, np=1, N=3

```
root@GAIA:/workspace/MPIJOBS/WJY/mpi_matrix_multiplication# mpiru
n -np 1 mpi_gatherv_test.o
It will compute A * x = y
A will be N by N Matrix, x and y will be N by 1 Matrix.
Input N
3
rank = 0, N = 3, N fix = 3, rows p = 3
```

$$\begin{bmatrix}9 & 2 & 6\\ 8 & 7 & 2\\ 8 & 0 & 9\end{bmatrix}\begin{bmatrix}6\\ 0\\ 1\end{bmatrix}=\begin{bmatrix}60\\ 50\\ 57\end{bmatrix}$$

```
= 0
rank = 0, A[0,0] = 9, A[0,1] = 2, A[0,2] = 6, x[0] = 6, y[0] = 60
, A[1,0] = 8, A[1,1] = 7, A[1,2] = 2, x[1] = 0, y[1] = 50, A[2,0]
 = 8, A[2,1] = 0, A[2,2] = 9, x[2] = 1, y[2] = 57,
```

Matlab

```
>> a = [ 9 2 6 ; 8 7 2 ; 8 0 9 ] ; x = [ 6 ; 0 ; 1]; a*x

ans =

   60
   50
   57
```

Validate Result

Code, np=2, N=3

```
root@GAIA:/workspace/MPIJOBS/WJY/mpi_matrix_multiplication# mpiru
n -np 2 mpi_gatherv_test.o
It will compute A * x = y
A will be N by N Matrix, x and y will be N by 1 Matrix.
Input N
3
rank = 0, N = 3, N_fix = 4, rows_p = 2
rank = 0, recvcounts[0] = 2, displs[0] = 0
rank = 0, recvcounts[1] = 1, displs[1] = 2
rank = 1, N = 3, N_fix = 4, rows_p = 1
```

$$\begin{bmatrix} 8 & 2 & 3 \\ 9 & 7 & 0 \\ 9 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 7 \\ 6 \end{bmatrix} = \begin{bmatrix} 48 \\ 67 \\ 66 \end{bmatrix}$$

```
= 0
= 2
rank = 1, A[0,0] = 9, A[0,1] = 6, A[0,2] = 1, x[0] = 6, y[0] = 66
,
rank = 0, A[0,0] = 8, A[0,1] = 2, A[0,2] = 3, x[0] = 2, y[0] = 48
, A[1,0] = 9, A[1,1] = 7, A[1,2] = 0, x[1] = 7, y[1] = 67,
```

Matlab

```
>> a = [ 8 2 3 ; 9 7 0 ; 9 6 1 ] ; x = [ 2 ; 7 ; 6 ]; a*x

ans =

    48
    67
    66
```

Validate Result

Code, np=3, N=3

```
root@GAIA:/workspace/MPIJOBS/WJY/mpi_matrix_multiplication# mpirun -np 3 mpi_gatherv_test.o
It will compute A * x = y
A will be N by N Matrix, x and y will be N by 1 Matrix.
Input N
3
rank = 2, N = 3, N_fix = 3, rows_p = 1
rank = 2, recvcounts[0] = 1, displs[0] = 0
rank = 2, recvcounts[1] = 1, displs[1] = 1
rank = 2, recvcounts[2] = 1, displs[2] = 2
rank = 0, N = 3, N_fix = 3, rows_p = 1
```

$$\begin{bmatrix} 8 & 3 & 0 \\ 9 & 1 & 0 \\ 6 & 4 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 42 \\ 33 \\ 42 \end{bmatrix}$$

```
= 0
= 1
= 2
rank = 2, A[0,0] = 6, A[0,1] = 4, A[0,2] = 0, x[0] = 4, y[0] = 42
,
rank = 1, A[0,0] = 9, A[0,1] = 1, A[0,2] = 0, x[0] = 6, y[0] = 33
,
Total Excute Time = 0.000014 seconds
rank = 0, A[0,0] = 8, A[0,1] = 3, A[0,2] = 0, x[0] = 3, y[0] = 42
```

Matlab

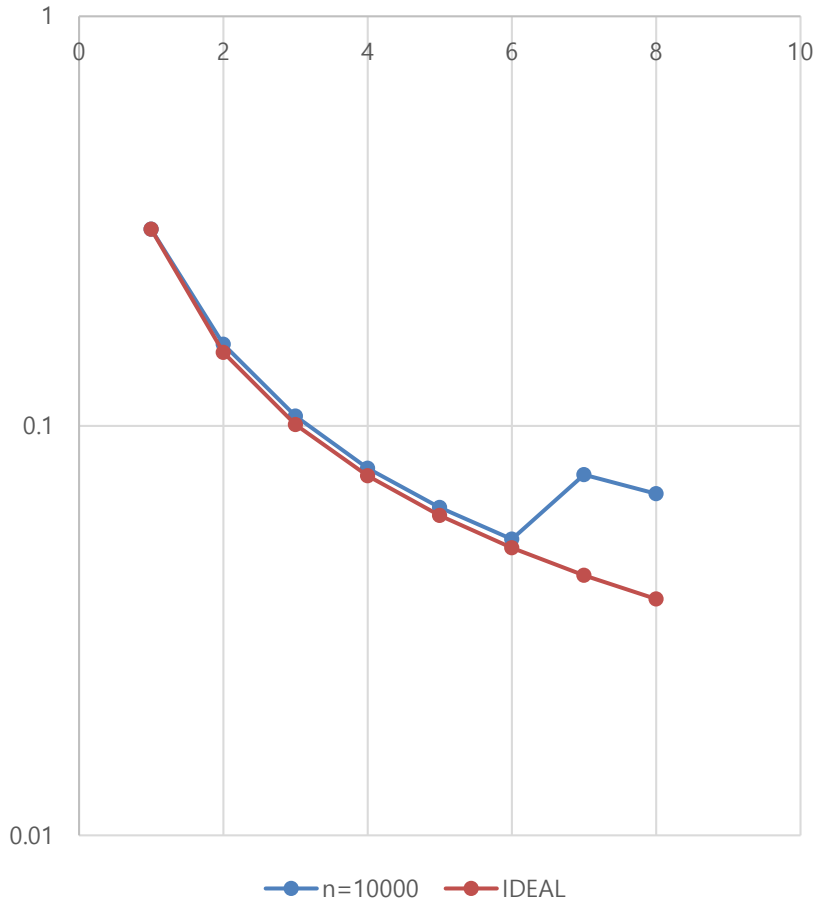
```
>> a = [ 8 3 0 ; 9 1 0 ; 6 4 0 ] ; x = [ 3 ; 6 ; 4 ]; a*x

ans =

    42
    33
    42
```

Result

Log(Time) vs NP



Speedup