Project 2 - Group 5

Team Members

Group Member	R#
Michael Beebe	R11772231
Diego Salas Noain	R11794236
Bandar Alkhalil	R11836831
Yongjian Zhao	R11915830
Denish Otieno	R11743138
Shiva Kumar Neekishetty	R11842757

Required Software

- MPI implementation (we are using Open MPI)
- C Compiler (such as gcc or clang)
- Make
- Bash

Instructions

Compile

To change the MPI wrapper to something other than `mpicc` (such as `mpich`), edit line 1 of the Makefile.

```
make
```

Run

```
./run.sh <desired number of processes>
```

The default number of processes is 4. You can change this by passing a command line argument when executing `run.sh`.

If you get an error saying "permission denied", run

```
chmod +x run.sh
```

then rerun `./run.sh`

Clean Build

```
make clean
```

Code Breakdown

The idea is that we divide processors into different row_group/col_group according to the rank. Then for each iteration k, we broadcast the corresponding row/col within its row_group/col_group, then calculate the min value.

generateRandomElements()

- Generate random elements for each submatrix
- The submatrix on diagonal (row_rank == col_rank) will be 0
- Other values will be random number between 1-9, or a large number INTINFINITY

print_subMatrix()

- Print value of each sub matrix

shortestPath()

- For each k, determine row_handeler and col_handler (k/chunk) which represents the row/col needs to be broadcast to rowgroups/colgroups
- Broadcast the corresponding row/col within rowgroup/colgroup

main()

- Initialize MPI.
- Initialize the W0 buffer.
- Initialize the W buffer.
- Split the main communication into two groups (for rows and columns grouping).

```
//the following buffer holds the row/column number K to be broadcast
initialize row_no_K and col_no_K

V=16
N= 16*16
chunk = ((V)/(no_P_worldCome/sqrt(no_P_worldCome)))
loop  k=0 into V-1
{
    //get the rank which handle the Kth row and column
    row_handeler = k/chunk
    col_handeler = k/chunk

    //update Kth row if this row assigned to you
    if (row_rank == row_handeler)
    {
        row_no_K = &W0[k%chunk]
    }

    //update Kth column if this row assigned to you
    if (col_rank == col_handeler)
    {
        loop from i=0 into chunk-1
        {
            col_no_K[i] = W0[i][k%chunk];
        }
    }

    //Broadcast Kth row and column to designated group
    broadcast row_no_K by row_handeler into row_group
    broadcat col_no_K by col_handeler into column_group

    loop from i=0 into chunk-1
    {
        loop from j=0 into chunk-1
        {
            W[i][j]= MIN(W0[i][j],(row_no_K[i] + col_no_K[j]));
            W0[i][j]=W[i][j];
        }
    }
}
```