



(../index.php)

**CS 475: Parallel
Programming**
Computer Science
Department
(<http://www.cs.colostate.edu>)
Fall 2016
Assignment 4 - Jacobi2D
MPI Blocking



(<http://welcome.colostate.edu/>)

Introduction

The objective of this assignment is two fold.

1. Solidify the concepts needed to distribute data between PEs corresponding to "ghost" or "halo" cells of a 2D stencil computation. In general, this is a tunable parameter, but for this assignment we want you to use the simplest halo: width 1.
2. Develop an intuition for the communication overhead in an MPI program.

To achieve this you will perform a series of programming tasks followed by a series of experimental tasks. The coding is quite tricky. I suggest you follow the steps outlined below. You will be implementing the communication required for a 1D data decomposition of Jacobi2D in MPI. The vast majority of the code is supplied. There are comments in the code in the locations that you need to update. The data domain is square (-p problemSize). Your block may or may not be square, but each PE only gets a single block. This means that there is a mathematical relationship between the problem size, the PE count and the block sizes (-x xsize -y ysize). You need to figure that out (and put it in your report).

Programming Tasks

1. **Vertical communication: [Jacobi2D-BlockParallel-MPI-VERT.test.c]** .
Insert the code needed to exchange ghost cells between neighbors to the north and south of each block. Remember to be cognizant of the order of your sends and receives as well as making sure you don't attempt to send to a rank that does not exist (for instance -1).
2. **Vertical & Horizontal communication: [Extra-credit] [Jacobi2D-BlockParallel-MPI.test.c]** The given code can handle both vertical and horizontal communication. Make appropriate changes to accept both horizontal and vertical tile sizes. Insert the code needed to exchange ghost cells between neighbors to the east and the west of each block. Keep in mind the same gotchas that you ran into for the north and south neighbors.
In addition to the above communication step, you will need to write the code

needed to pack/unpack a column of a 2D array as implemented in the code into a single vector. There are empty functions (packColToVec and unpackVecToCol) at the top of the file for you to use for this purpose. You will need to call the pack function before MPI_Send, and the unpack function after MPI_Recv for the east-to-west communication.

Experimental Tasks

1. Vertical communication: You should come up with a hypothesis about which tile size is going to perform best for this algorithm and test that hypothesis. Your hypothesis should be the first section of your report. Run your experiments for the following problem size on 1-64 processors of the **Cray** and report where your code stops scaling. Make sure not to use more than 2 nodes. You will need to note the data footprint and the cache size when allocating nodes and PEs.
 1. -p 10000, -T 50.
 2. -p 50000, -T 20.
2. Horizontal communication **[Extra-credit]**: Use the same two problem instances as you used for vertical communication for your experimental evaluation. Here, you will have a choice to explore -x xsize and -y ysize for a given problem size and number of processors.

Code Submission

Submit your PA4.(1/2).tar containing vertical communication file (Jacobi2D-BlockParallel-MPI-VERT.test.c). You may also include the extra-credit file Jacobi2D-BlockParallel-MPI.test.c.

Report Submission

Outline

1. Hypothesis
2. Algorithm Description
3. Blocking description: include details on ghost cell exchanges.
4. Experimental Setup
5. Results
6. Conclusions and Analysis (was your hypothesis correct?).

Some Notes on Getting Started

Download the Starter code (PA4.tar). You can download the **simpler version** PA4_simple (PA4_simple.tar).

The provided MPI code is complete for sequential execution. It also has a validation code. When you open the tarball attempt to make the executable by typing the following:

```
make Jacobi2D-BlockParallel-MPI
```

Then attempt to run the code and make sure that it validates for serial execution (number of processor=1)

On Cray (required to run only on Cray)

```
$ aprun -n1 Jacobi2D-BlockParallel-MPI -p 8 -x 8 -y 8 -v
```

```
Time: 0.000019
```

```
SUCCESS
```

On CS machines (can run on CS machines to debug code before transferring files)

```
$ mpirun -np 1 Jacobi2D-BlockParallel-MPI -p 8 -x 8 -y 8 -v
```

```
Time:0.000034
```

```
SUCCESS
```

Note the command line arguments.

p is the problem size.

x is the length of one tile in the Horizontal direction.

y is the length of one tile in the Vertical direction.

v indicates that validation should take place.

Do NOT change the command line argument parsing code (in util.h).

DO NOT run your parallel code with validation flag.

You can see the Grading Rubric (PA4_Grading_Rubric.pdf) here.

Session Time 1001

Secs.

Originating IP

123.230.131.249

User: Guest (../..

/ztools

/authenticateLogin.php)

Apply to CSU (<http://admissions.colostate.edu>)

Contact CSU (<http://www.colostate.edu/info->

[contact.aspx](http://www.colostate.edu/info-contact.aspx)) | Disclaimer

(<http://www.colostate.edu/info-disclaimer.aspx>)

Equal Opportunity (<http://www.colostate.edu/info->

[equalop.aspx](http://www.colostate.edu/info-equalop.aspx))

Colorado State University, Fort Collins, CO 80523

USA

© 2016 Colorado State University

