ME471/571 - Example problem 1

Parallel standard deviation program using collective communication

Michal A. Kopera

Due date: Feb. 8th, 3 pm

This is an example problem, similar to what you can expect on a project. You will work on it in class, and possibly finish as a homework for the week.

In this problem I ask you to write a parallel version of a program which computes standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(x_i - \mu)^2}, \qquad \text{where} \qquad \mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i$$

You will test the program on a dataset containing values of height (in inches) for a representative sample of men and women in the U.S. To begin this exercise, log in to R2 and execute the following command:

```
cp /home/makopera/scratch/std_dev_exercise .
```

to get the necessary files. You have the same files provided in Google Drive, but it may take a while to upload them to r2, as the data files are quite big. The package contains:

data_men.csv - a file with height data for about 1.5 min U.S. men,

data_women.csv - a file with height data for about 1.5 min U.S. women,

data_test.csv - a sample data with only 16 entries to use when developing the code.

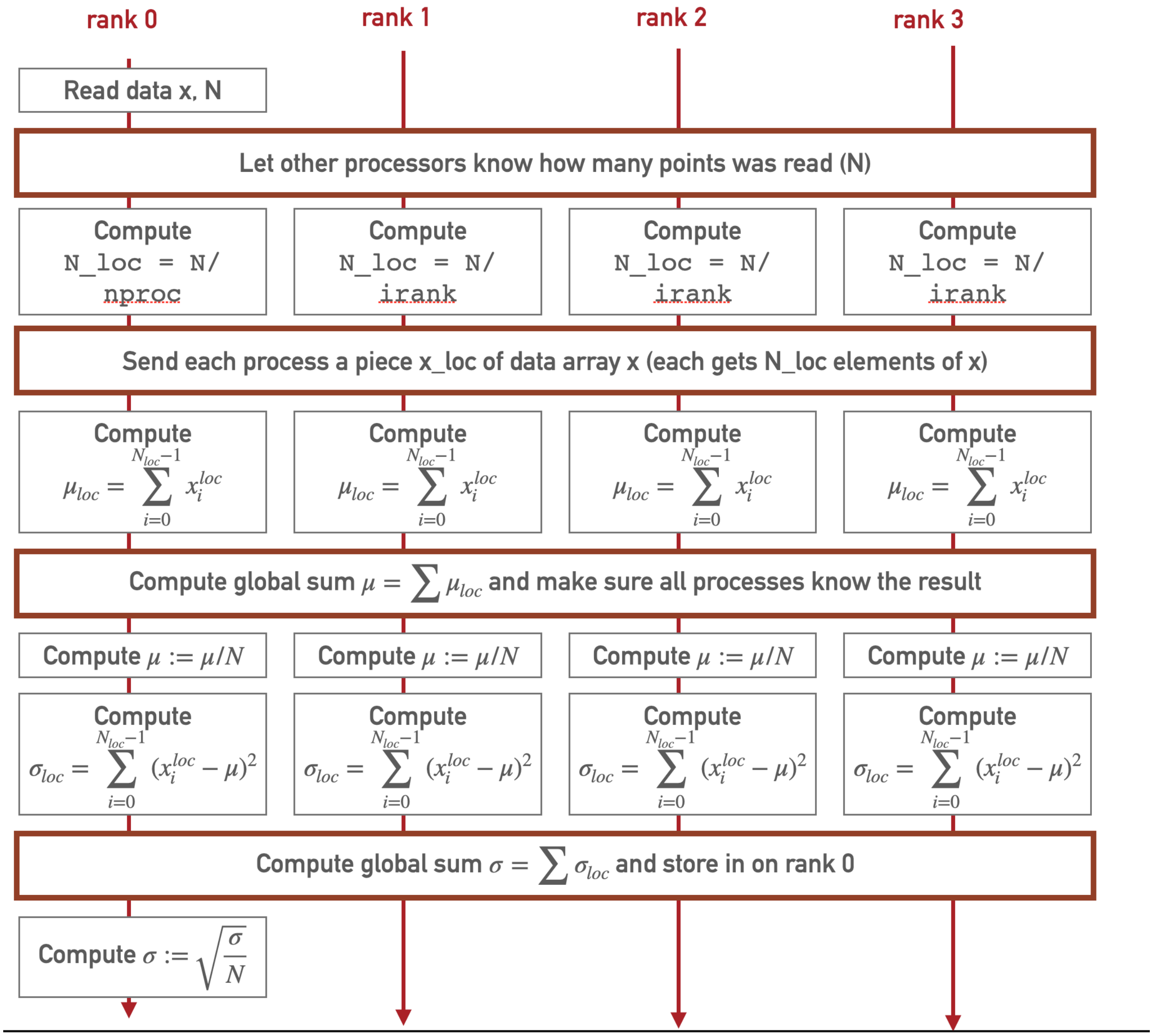
In each data file the first number is the total number of data entries, so for the data_test.csv it is 16, and for data_women.csv it is 1569642.

The package also contains the code std_dev_mpi.c, Makefile and a run script. Complete the following tasks.

Task 1

The code in std_csv_mpi.c is only a serial code for computing standard deviation of a data set, with some missing elements to make it parallel. Use the Jamboard we have worked on last time to help you identify missing pieces. The code has MPI_Init() and MPI_Finalize(), but nothing more than that. Use the data_test.csv file when you test the code to avoid working with a large data set.

The parallel algorithm we have worked with so far looks as follows:


rank 0
rank 1
rank 2
rank 3
Read data x, N
Let other processors know how many points was read (N)
Compute N_loc = N/ nproc
Compute N_loc = N/ irank
Compute N_loc = N/ irank
Compute N_loc = N/ irank
Send each process a piece x_loc of data array x (each gets N_loc elements of x)
Compute $\mu_{loc} = \sum_{i=0}^{N_{loc}-1} x_i^{loc}$
Compute $\mu_{loc} = \sum_{i=0}^{N_{loc}-1} x_i^{loc}$
Compute $\mu_{loc} = \sum_{i=0}^{N_{loc}-1} x_i^{loc}$
Compute $\mu_{loc} = \sum_{i=0}^{N_{loc}-1} x_i^{loc}$
Compute global sum $\mu = \sum \mu_{loc}$ and make sure all processes know the result
Compute $\mu := \mu/N$
Compute $\mu := \mu/N$
Compute $\mu := \mu/N$
Compute $\mu := \mu/N$
Compute $\sigma_{loc} = \sum_{i=0}^{N_{loc}-1} (x_i^{loc} - \mu)^2$
Compute $\sigma_{loc} = \sum_{i=0}^{N_{loc}-1} (x_i^{loc} - \mu)^2$
Compute $\sigma_{loc} = \sum_{i=0}^{N_{loc}-1} (x_i^{loc} - \mu)^2$
Compute $\sigma_{loc} = \sum_{i=0}^{N_{loc}-1} (x_i^{loc} - \mu)^2$
Compute global sum $\sigma = \sum \sigma_{loc}$ and store in on rank 0
Compute $\sigma := \sqrt{\frac{\sigma}{N}}$

And you can use attached MPI handout to find out the syntax for the collective communication operations: Broadcast, Reduce, Gather, Scatter.

Test your code on the data_test.csv using different number of processes to make sure you always get the same answer.

Task 2

Once the program is complete and you get the same result of standard deviation for the test data regardless of how many processes you use, pick one of the data sets (men or women) and compute the standard deviation and mean height for selected group. You should get answers similar to:

$$\mu_{men} = 70, \sigma_{men} = 3.0$$

$$\mu_{women} = 65, \sigma_{women} = 2.5$$

Task 3

Once you are certain you are getting correct results, time the code for n=1, 2, 4, 8 processors and plot the time it takes to run the program against the number of processes. How does the time depend on n?

Mastery

Can you measure how much time the code spends in computation only, and how long does it take to communicate? You can use MPI_Wtime() to help you measure different parts of the code, and add up appropriate times together.