Project 2, due on 04/22.

Problem 1 for undergraduate students. Compute Ax+b

Implement a parallel algorithm to compute **Ax+b,** in which **A** is a matrix, **x** and **b** are vectors. Use the base code saved under directory ~zxu2/Public/ACMS40212/col_decomp_mat_vec_multi to start your implementation.

Generate a 4112×2056 matrix, a vector **x** of size 2056 and vector **b** of size 4112 respectively to test your result. The base code which needs to be modified to generate the data is at:

~zxu2/Public/ACMS40212/col_decomp_mat_vec_multi/data_gen/myio.c

Use 2, 4, and 8 nodes to run the program respectively. This project must be done individually.

Hints:

1. Modify my_io.c to generate one matrix and two vector data files, respectively.

2. The current parallel matrix-vector multiplication code using column-wise decomposition utilizes point-to-point communication to implement functions reading in the vector data from a file and printing out the result respectively. See C functions read_block_vector() and print_block_vector(). Rewrite these two functions to use proper collective communication to implement the same functionality respectively and compute **Ax+b.**

The available collective communication functions are: MPI_Scatter(), MPI_Scatterv(), MPI_Gather(),MPI_Gatherv(), etc. Which ones to use?

3. **y**=**Ax** is computed using column-wise decomposition. Matrix **A** is read in by function read_col_striped_matrix(); while vector **x** is read in by function read_block_vector(). Use the row-wise decomposition of vectors **y** and **b** to compute the final answer. This requires to implement a new function similar to read_block_vector() to read in the vector **b**. Name this new function read_block_vector_b().

4. Validate your result by computing L1 norm of the solution and compare this with the result obtained by a serial code (You can modify myio.c for this purpose).

Hand-In.

1. Send the source code to me by email. Please use the email title: acms40212S14-Proj2-your-ND-ID.

2. A report which contains results and a description of your algorithm using the pseudo code language. You need to explain which MPI collective communication function is used and how it is used.

Problem 2 for graduate students. Compute Ax+b. Using Process Topology to Implement.

Implement a parallel algorithm to compute **Ax+b,** in which **A** is a matrix, **x** and **b** are vectors. Use 2D block decomposition and 2D grid topology of the process to implement this computation in parallel.

The base code is at the folder:

~zxu2/Public/ACMS40212/2D_decomp_mat_vec_multi

Generate a 4112×2056 matrix, a vector **x** of size 2056 and vector **b** of size 4112 respectively to test your result. The base code which needs to be modified to generate the data is at:

~zxu2/Public/ACMS40212/col_decomp_mat_vec_multi/data_gen/myio.c

Use 3×2 and 4×2 nodes to run the program respectively. This project must be done individually.

Hints:

1. When distributing the vector **x** among processors, implement the algorithm shown in Figure (b) on page 22 of lecture notes "**Parallel matrix algorithms (part 2)**".

Use row communicators and column communicators to scatter and broadcast the vector.

2. Gather the result of computing **y**=**Ax** to processes in the first column of the process grid. For example, if we assume a fine-grained 2D decomposition as shown by the picture on page 19 of lecture notes "**Parallel matrix algorithms (part 2)**", where each entry of the 4×4 matrix is assigned to a process, solution y_0 is stored on process P_0, y_1 is stored on process P_4, y_2 is stored on process P_8, and y_3 is stored on process P_{12}, respectively.

3. Distributing vector **b** only among processes in the first column of the process grid such that only these processes compute **y+b**.

4. Validate your result by computing L1 norm of the solution and compare this with the result obtained by a serial code (You can modify myio.c for this purpose).

Hand-In.

1. Send the source code to me by email. Please use the email title: acms40212S14-Proj2-your-ND-ID.

2. A report which contains results and a description of your algorithm using the pseudo code language. You need to explain which MPI collective communication function is used and how it is used.

.