Designing and building applications for extreme scale systems
CS598 SP2016
HW8: Communication Performance with MPI

Goals

- Gain experience with MPI
- Explore communication performance models

Tasks

In this exercise you will study the performance of MPI communication in several different situations. This will also give you an opportunity to learn how to run MPI programs in your environment.

Write an MPI program that performs the following 2 communication performance tests. In each case, you will time the performance of the communication using MPI_Wtime.

1. Process i sends a message of size m to process j, then receives a message of the same size from process j (this is a pingpong test). See below for i, j, and m.
    a. Use MPI_Send and MPI_Recv
    b. Use MPI_Isend, MPI_Irecv, and MPI_Wait

2. Process i and process j send messages of size m to each other, then receive a message of size m from each other (this is the "head-to-head" variant of the pingpong test). You must (make sure you understand why you must) use nonblocking communication here. See below for i, j, and m.
    a. Use MPI_Isend, MPI_Irecv, and MPI_Wait

Use the following 3 test cases. wsize is the number of processes in MPI_COMM_WORLD. Make sure that you run on at least two nodes; the easiest way to do that is to place only one process per node. Use at least 4 processes; if you are using Blue Waters, use 32 processes (and hence 32 nodes).

1. i = 0, j = 1; m = 0, 4, 8, …, 262144 bytes (i.e., 0 bytes and 2^k bytes for k=2,…,18). Other processes are idle.
2. i = 0, j = wsize - 1; m = same as the previous test. Other processes are idle.
3. i = any process with even rank in MPI_COMM_WORLD, j = i+1. m = same as the previous test.

When you measure time for these operations, ensure that you (a) measure a long enough time to be significant (use MPI_Wtick to check the resolution of MPI_Wtime) and (b) make at least 5 separate measurements for each test. Report the minimum time taken for the 5 measurements (this is the most likely to be repeatable).

Plot the time for communication of a single exchange (send and receive) as a function of message size for each case (there are 9 different cases). Use a single graph (i.e., all data on the same graph, so that you can see the relationships between the 9 different cases).

Analyze the data by answering the following questions:

1. Compare the time for case 1 and case 2 (that is, communication between ranks 0 and 1 and between ranks 0 and wsize-1). Are they the same within experimental error? If not, can you suggest why that may be?
2. For case 1 (communication between ranks 0 and 1), and the pingpoing pattern, compute a fit to the communication time using the simple performance model T = s + rn, where n is the number of bytes. Because it is very likely that 2 or even three different approaches are used by the MPI implementation to send the data (based on the message size), you may need to divide the data into several subsets:
    a. If there is an obvious change in performance at a relatively large message size (8k on Blue Waters, for example), fit the model T=s+rn for data sizes between 8 bytes and the size before the change (e.g., 8, 16, 32, …, 4096 on Blue Waters), and separately for sizes starting at the point where there is the performance change (8192, …, 262144 bytes on Blue Waters).
    b. If there is no obvious change in performance, fit all of the data for sizes from 8 bytes to 262144 bytes.

    In either of the above cases, report the values of s and r for a single communication exchange. You can use a simple least squares fit to compute the parameters. Plot the model and the data on the same graph to show the fit to the data.

Submit

A PDF file containing your MPI program, and table & plots (there should be at least two plots – one for the timing results for the 9 tests, and one showing the fit for the pingpong communication case with i=0 and j=1) for the results of the tests, including a table giving the values of s and r determined by a least squares fit described above. A brief discussion is necessary: do the results match your expectations?

To think about (but not turn in)

1. Why does the analysis above start from 8 bytes instead of 0, since you have data a 0 and 4 bytes? How would your results change if you included that data?
2. How does the performance of a single pair of processes communicating compare to all processes communicating? What does the simple model predict? If they are different, what do you think might be happening? How could you test your hypothesis?

3. Is there an extra cost for using nonblocking communication (i.e., s is larger)? Is the value of r consistent for blocking and nonblocking communication?
4. Analyze the other communication patterns and for each of the 3 test cases, obtaining a least squares fit for the model T=s + rn. What values of s and r do you get? Do the values match your expectations?