**Learning Objectives**

• To study the principles of message passing

• To learn the message passing interface (MPI)

Write an MPI program (i.e., parraysum.c) running on the DMC cluster of the Alabama Supercomputer Center. You may modify the source code that you have studied in the above step. The suggested algorithm is given as follows:

1. Node 0 reads in the array
2. Node 0 scatters a fragment of the array to every slave Node
3. Each slave node sums the array fragment received from Node 0
4. Node0 collects the sums from all the slave nodes and prints the final result

You may apply one of the following approach to making Node 0 calculate the final result. First, Node0 may use the MPI\_SUM reduction to sum the sums from the slave nodes. Second, Node 0 may use the gather operation to gather the sums offered by the slave nodes.

* You must implement the parallel algorithm in the sumArray() function. You will lose points if this function is not correctly implemented in your MPI program.
* Your parallel program must evenly distribute load across N nodes. For example, given N Nodes and M array elements, each node should sum at least M / N elements. Of course, one node (e.g., Node 0) should handle the remaining M % N elements.
* Node 0 in your program not only needs to display the sum of the array, but also has to show the execution time of your parallel program.
* Your MPI program must take an input file as a command line argument. For example, you may run your program as:

>mpirun –np 4 parraysum data1000.txt