|  |  |
| --- | --- |
| avg\_rows\_per\_process = num\_rows / num\_procs;  if(my\_id == root\_process) {  /\* distribute a portion of the vector to each child process \*/    **for(an\_id = 1; an\_id < num\_procs; an\_id++) {**  **start\_row=an\_id\*avg\_rows\_per\_process + 1;**  **end\_row=(an\_i+1)\*avg\_rows\_per\_process;**  if((num\_rows - end\_row) < avg\_rows\_per\_process)  end\_row = num\_rows - 1;  num\_rows\_to\_send = end\_row - start\_row + 1;  **ierr = MPI\_Send(&num\_rows\_to\_send, 1 , MPI\_INT,an\_id, send\_data\_tag, MPI\_COMM\_WORLD);**  **ierr = MPI\_Send(&array[start\_row], num\_rows\_to\_send, MPI\_INT,an\_id, send\_data\_tag, MPI\_COMM\_WORLD);**  }  /\* and calculate the sum of the values in the segment assigned to the root process \*/  **sum = 0;**  **for(i = 0; i < avg\_rows\_per\_process + 1; i++) {**  **sum += array[i];**  **}**  printf("sum %i calculated by root process\n", sum);  /\* and, finally, I collet the partial sums \*/  **for(an\_id = 1; an\_id < num\_procs; an\_id++) {**  **ierr = MPI\_Recv( &partial\_sum, 1, MPI\_LONG,MPI\_ANY\_SOURCE,return\_data\_tag, MPI\_COMM\_WORLD, &status);**  **sender = status.MPI\_SOURCE;**  **printf("Partial sum %i returned from process %i\n", partial\_sum, sender);**  **sum += partial\_sum;**  **}**  printf("The grand total is: %i\n", sum);} | else {  /\* slave process, \*/  **ierr = MPI\_Recv( &num\_rows\_to\_receive, 1, MPI\_INT,**  **root\_process, send\_data\_tag, MPI\_COMM\_WORLD, &status);**    **ierr = MPI\_Recv(&array2, num\_rows\_to\_receive, MPI\_INT, root\_process, send\_data\_tag, MPI\_COMM\_WORLD, &status);**    num\_rows\_received = num\_rows\_to\_receive;  /\* Calculate the sum of my portion of the array \*/  partial\_sum = 0;  **for(i = 0; i < num\_rows\_received; i++) {**  partial\_sum += array2[i];  **}**  /\* and finally, send my partial sum to the root process \*/    **ierr = MPI\_Send( &partial\_sum, 1, MPI\_LONG, root\_process,return\_data\_tag, MPI\_COMM\_WORLD);**  } |