

Possible Questions to evaluate students knowledge of Learning Objectives:

•**APPLY** basic MPI functions to accelerate a common scientific pattern

Sample Questions:

1. Rewrite provided code provided so instead of using only the neighbor to the North, South, East, and West uses also the neighbor

NW	North	NE
West	Center	East
SW	South	SE

2. Ask students to change the Heat parameters like SPEED, Increase/decrease the number of iterations for the stencil, and make them explain the difference in results

•**USE** Profiler tools to evaluate bottlenecks in code

1. Use the function MPI_Wtime() to measure execution times of different loops and/or functions on the code. Example

```
double start = MPI_Wtime();  
  
// Do something here lets call it ABC  
  
double end = MPI_Wtime();  
  
printf("ABC took %1.2f seconds to run\n", end-start);
```

•**IDENTIFY** which loops/task can be distributed (loop dependency)

1. Does the next loop contain a loop dependency?

```
for( i = 1; i<100; i++){  
    A(i) = B(i)+1  
    C(i) = A(i-1)*2    <- yes here, to calculate this iteration we need the prev. one  
}
```

2. Does the next loop contain a loop dependency?

```
for( i = 1; i<100; i++){
```

```

        C(i) = A(i-1)*2    <- yes here, to calculate this iteration we need the prev. one
    }

```

3. Does the next loop contain a loop dependency?

```

    for( i = 1; i<100; i++){
        A(i) = B(i-1)*2    <- no
    }

```

•**COMPARE** sequential to distributed execution times

1. Ask students to change the number of MPI processes and obtain runtimes of sequential code (also provided) vs. distributed code using those different process
Create a table and graphs to study execution

Sequential	
MPI code 1 process	
MPI code 2 process	
....	

Advanced Questions:

•(code) The stencil code :

Rewrite code for stencil so instead of hardcoding in the loop the neighbor cells, make the code more flexible and take this neighborhood and its coefficients as a filter passed in one input parameter

Note: The goal of this question is to show that the stencil code is really a filter operation and is applicable to many different problems and disciplines. A follow up question is to implement a Gaussian Blur for images reusing the code just created

•(code) Change it to 3D stencil code

Note: The goal of this question is to show that the same code just developed can be very easily changed into a 3D input/output and therefore can be applied to for example diffusion of a liquid into another one with different physical properties

•(code) Chunk not just by rows do it also by column

Note: the goal of this question is to further accelerate the stencil code since it will make it more scalable.