# Assignment – 5

## Group number-29

1. Manvitha Aathmuri – R11847781
2. Raviteja Sirisanagandla - R11851152
3. Sai Tejaswini Chirumamilla - R11840694
4. Preetham Alwala - R11846544
5. Girish Gandham - R11870388
6. Ganesh Reddy Mannem - R11849144

## Problem Statement:

The problem is to parallelize the force calculation for a system of n particles, ensuring balanced computation loads across processes and achieving a communication cost complexity of O(n log_2 p), where n is the number of particles and p is the number of processors. The given serial code calculates the forces between particles based on their positions.

## Technical Approach:

The force computation problem for n particles is solved technically by using OpenMP directives for shared-memory parallelism. The main goal is loading balancing, which is accomplished by parallelizing the outer loop to divide the burden equally across the threads. To manage the summation of forces in a parallel setting, a reduction clause is employed, guaranteeing that every thread keeps a local copy of the force array (f). By doing this, congestion for shared resources is reduced and synchronization overhead is minimized. To avoid data races and preserve the accuracy of the parallel execution, private variables (i, j, diff, and tmp) are also established inside the parallel zone. In order to meet scalability objectives, the selected parallelization technique seeks to achieve a communication cost complexity of O(n log_2 p) This allows threads to work independently on their local data.

In order to ensure that force calculations are accurate and that the parallelization approach scales with an increasing number of processors, it is carefully constructed. By reducing communication overhead, the reduction clause improves the effectiveness of the parallel implementation. The methodology offers a basis for the effective parallelization of force computations in massive particle systems, and it can be further optimized according to particular hardware features and application needs.

## Conclusion:

The parallelization of the force calculation demonstrates significant improvements in execution time for large particle systems. Using OpenMP parallelism and cautious shared and private variable management, the task is split up among threads. The reduction clause prevents data races while guaranteeing the correct accumulation of forces. It is important to remember that depending on the hardware architecture and the unique properties of the particle system, different performance increases could be obtained. The suggested method offers a strong basis for parallelizing the force calculation in a shared-memory setting, but additional optimizations or alternative parallelization techniques may be investigated for various cases.

## Steps to Execute:

- Open the Group_29_assignment_5.c file's location in terminal using the below command:

  nano Group_29_assignment_5.c

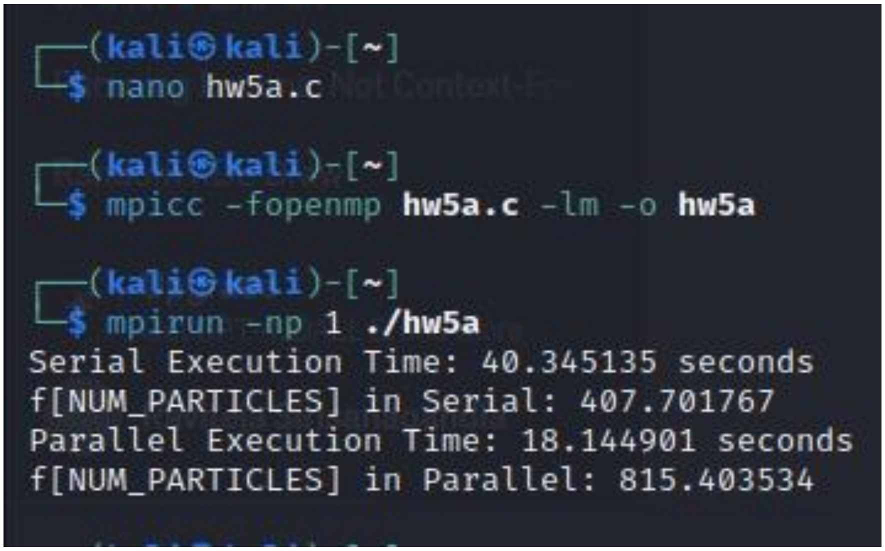
- Compile the file Group_29_assignment_5.c using the below command:

  mpicc -fopenmp Group_29_assignment_5 -lm -o Group_29_assignment_5.c

- Execute the file using below command for the output to be visible in console log:

  Mpirun -np 1 ./Group_29_assignment_4

## Output:



```
┌──(kali㉿kali)-[~]
└─$ nano hw5a.c

┌──(kali㉿kali)-[~]
└─$ mpicc -fopenmp hw5a.c -lm -o hw5a

┌──(kali㉿kali)-[~]
└─$ mpirun -np 1 ./hw5a
Serial Execution Time: 40.345135 seconds
f[NUM_PARTICLES] in Serial: 407.701767
Parallel Execution Time: 18.144901 seconds
f[NUM_PARTICLES] in Parallel: 815.403534
```