REPORT

Name - Bharti Verma
Roll No. - 102303575
Group - 3C42
Assignment - 2

Q1 Molecular Dynamics - Force Calculation

Aim :

To evaluate how OpenMP parallelization improves the execution speed of molecular force calculations.

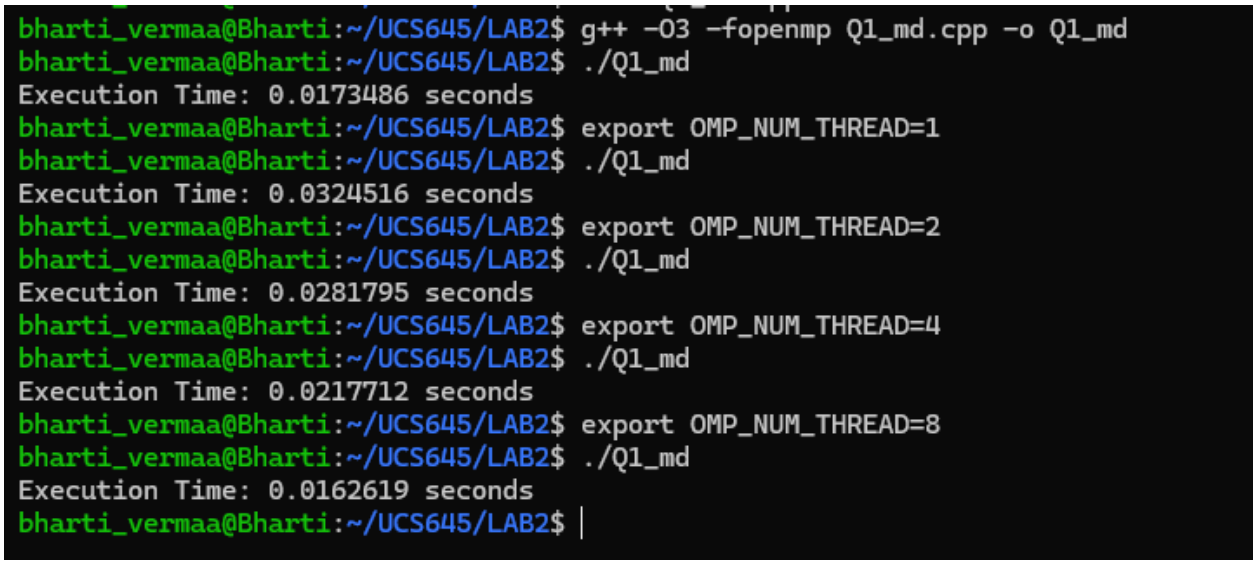
Problem Description

The program computes pairwise forces between particles, which is computationally expensive. The goal is to divide this workload across multiple CPU cores using OpenMP.

Methodology

The outer loop of the force computation was parallelized so that each thread handles a subset of particles independently.

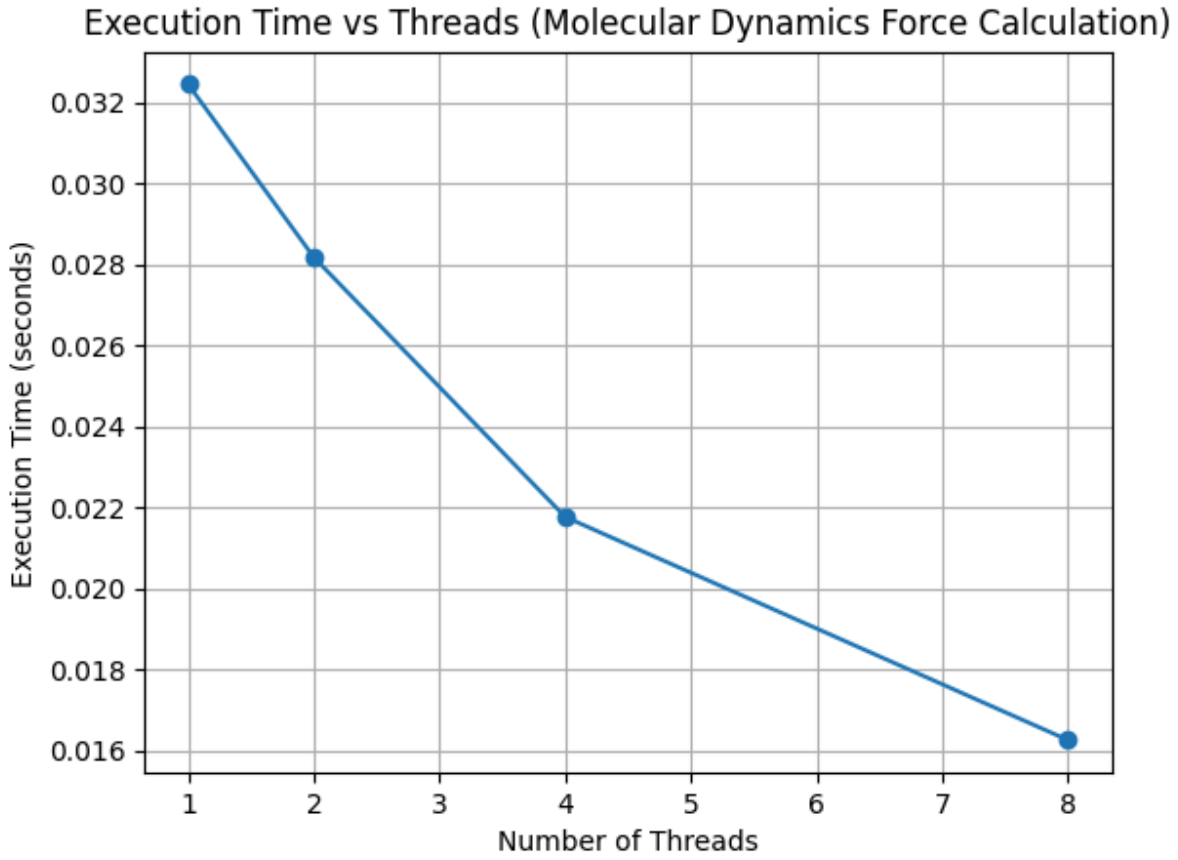
SnapShots



Results:

Threads	Time
1	0.0324516
2	0.0281795
4	0.0217712
8	0.0162619

Graph:



Observation:
Execution time decreases as the number of threads increases, indicating successful parallelization of the force computation.

Analysis:
The speedup is sub-linear and efficiency drops with higher thread counts due to synchronization overhead and shared memory access.

Memory Usage:
The algorithm uses shared particle position arrays with per-thread local force accumulation, leading to moderate but scalable memory usage.

Conclusion:
OpenMP parallelization improves performance of molecular dynamics force calculation, but scalability is limited by memory and parallel overheads.

Question 2: DNA Sequence Alignment

Aim
To study the behavior of the Smith–Waterman DNA alignment algorithm under parallel execution.

Problem Description
DNA alignment relies on dynamic programming where each computation depends on previous values, making parallelization difficult.

Methodology
Wavefront parallelization was used to execute independent diagonals of the alignment matrix using OpenMP.

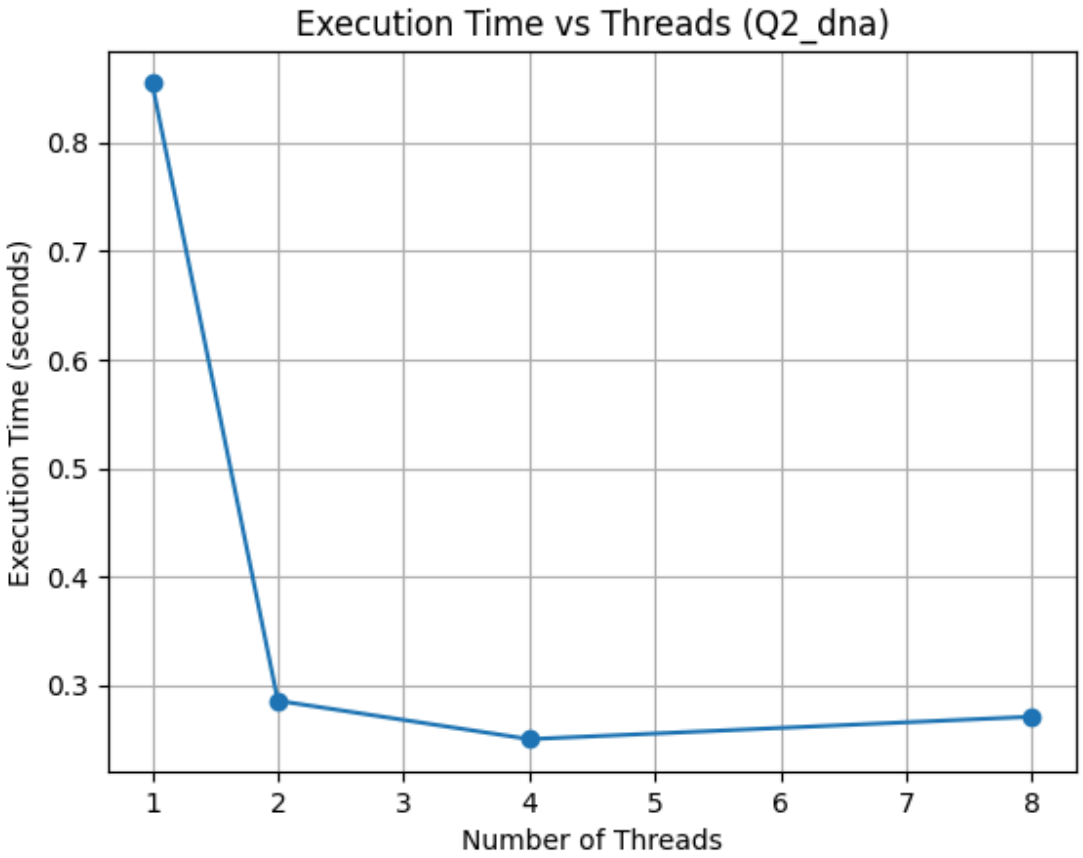
SnapShots

```
bharti_vermaa@Bharti:~/UCS645/LAB2$ g++ -O3 -fopenmp Q2_dna.cpp -o Q2_dna
bharti_vermaa@Bharti:~/UCS645/LAB2$ ./Q2_dna
Time: 0.236757 seconds
bharti_vermaa@Bharti:~/UCS645/LAB2$ export OMP_NUM_THREAD=1
bharti_vermaa@Bharti:~/UCS645/LAB2$ ./Q2_dna
Time: 0.854927 seconds
bharti_vermaa@Bharti:~/UCS645/LAB2$ export OMP_NUM_THREAD=2
bharti_vermaa@Bharti:~/UCS645/LAB2$ ./Q2_dna
Time: 0.285704 seconds
bharti_vermaa@Bharti:~/UCS645/LAB2$ export OMP_NUM_THREAD=4
bharti_vermaa@Bharti:~/UCS645/LAB2$ ./Q2_dna
Time: 0.250321 seconds
bharti_vermaa@Bharti:~/UCS645/LAB2$ export OMP_NUM_THREAD=8
bharti_vermaa@Bharti:~/UCS645/LAB2$ ./Q2_dna
Time: 0.271007 seconds
bharti_vermaa@Bharti:~/UCS645/LAB2$
```

Results

Threads	Time
1	0.854927
2	0.285704
3	0.250321
4	0.271007

Graphs:



Observation:

Execution time decreases significantly from 1 to 4 threads, with a slight increase at 8 threads.

Analysis:

The initial speedup indicates effective parallelization, while the slowdown at higher threads is due to parallel overhead and contention.

Memory Usage:

The program uses shared input data with per-thread computation, resulting in moderate and scalable memory usage.

Conclusion:

OpenMP improves performance for the DNA computation up to an optimal thread count, beyond which overhead limits scalability.

Question 3: Heat Diffusion Simulation

Aim

To analyze scalability of a grid-based heat diffusion simulation using OpenMP.

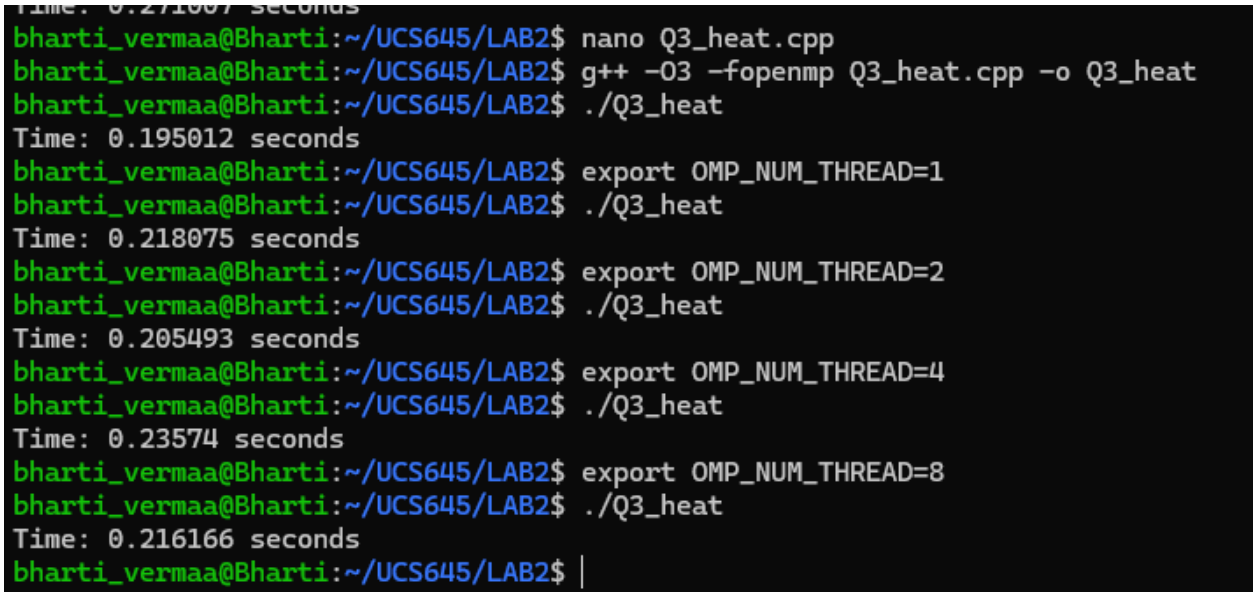
Problem Description

Each grid cell update depends only on its neighbors, making the task highly parallelizable.

Methodology

Nested loops were parallelized using OpenMP collapse directive to distribute iterations evenly.

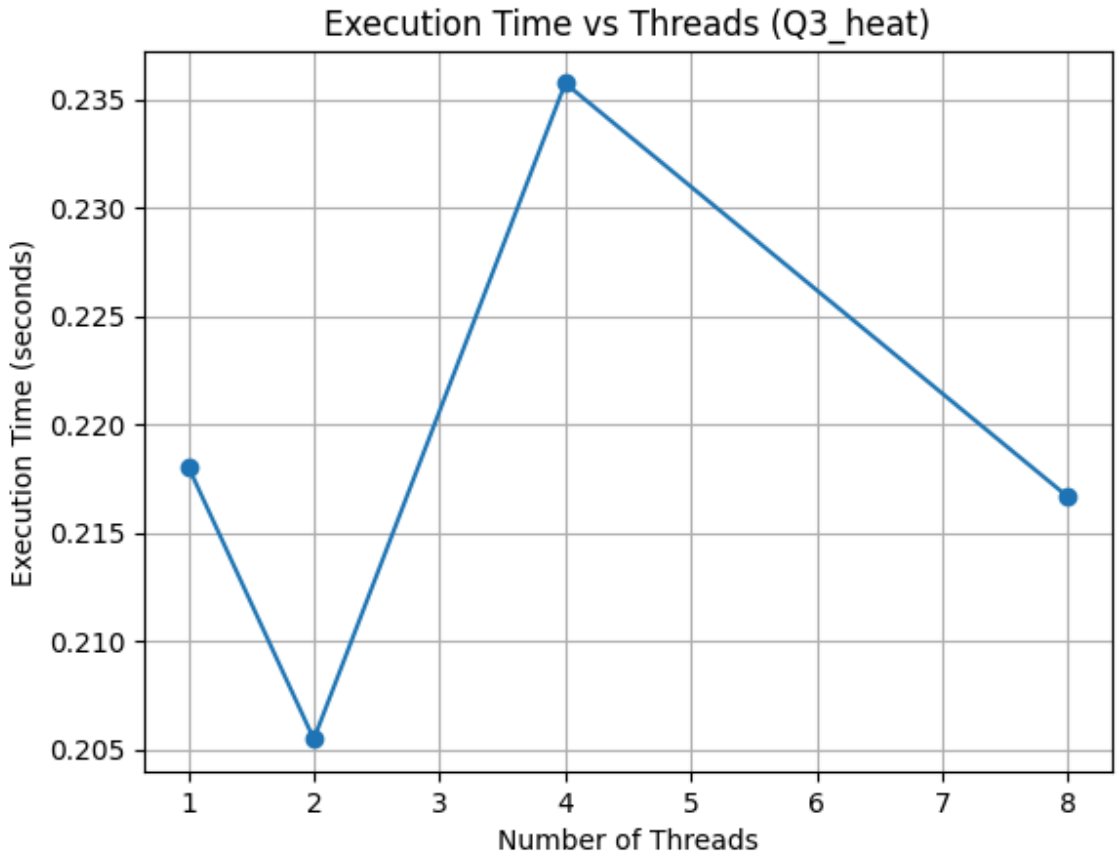
SnapShots



Results:

Thread	Time
1	0.218075
2	0.205493
4	0.235740
8	0.216660

Graphs:



Observation:

Execution time shows marginal improvement at 2 threads but fluctuates for higher thread counts.

Analysis:

Synchronization and memory access costs dominate the computation, reducing the benefits of additional threads.

Memory Usage:

The heat simulation relies on shared grid data, leading to increased memory traffic and limited cache reuse.

Conclusion:

The heat computation exhibits limited scalability, as memory bandwidth and synchronization overhead constrain parallel performance.