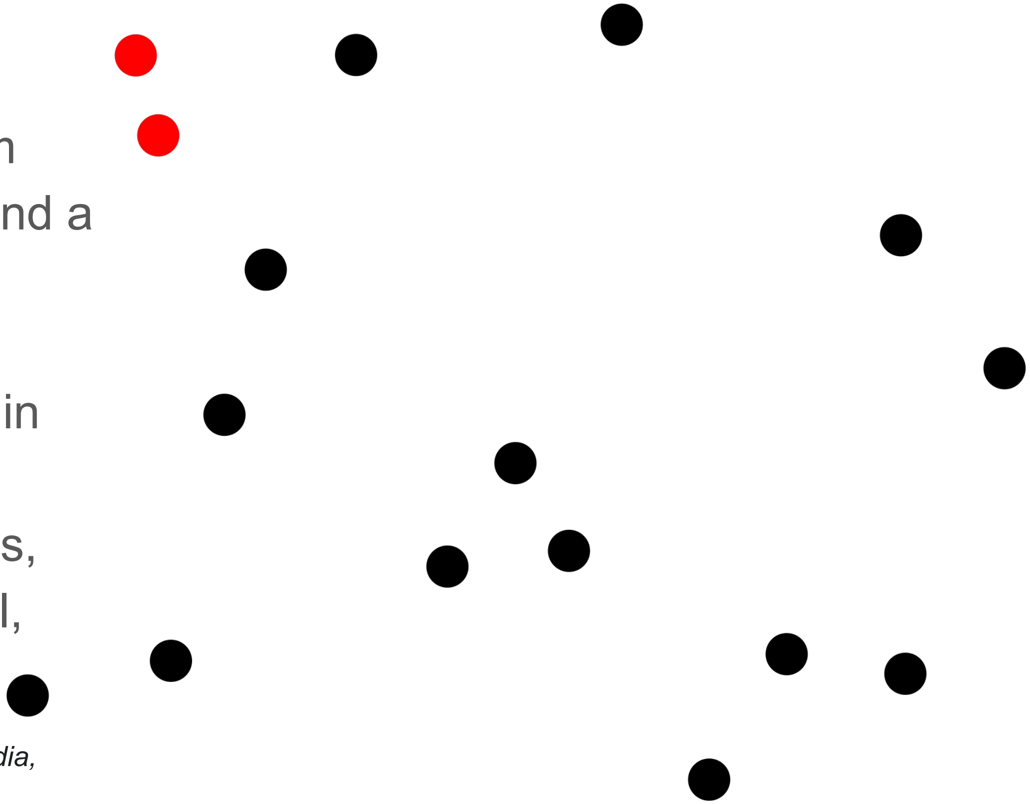
Closest pair of points problem

Sai **Praveen** Mylavarapu

May 9, 2023

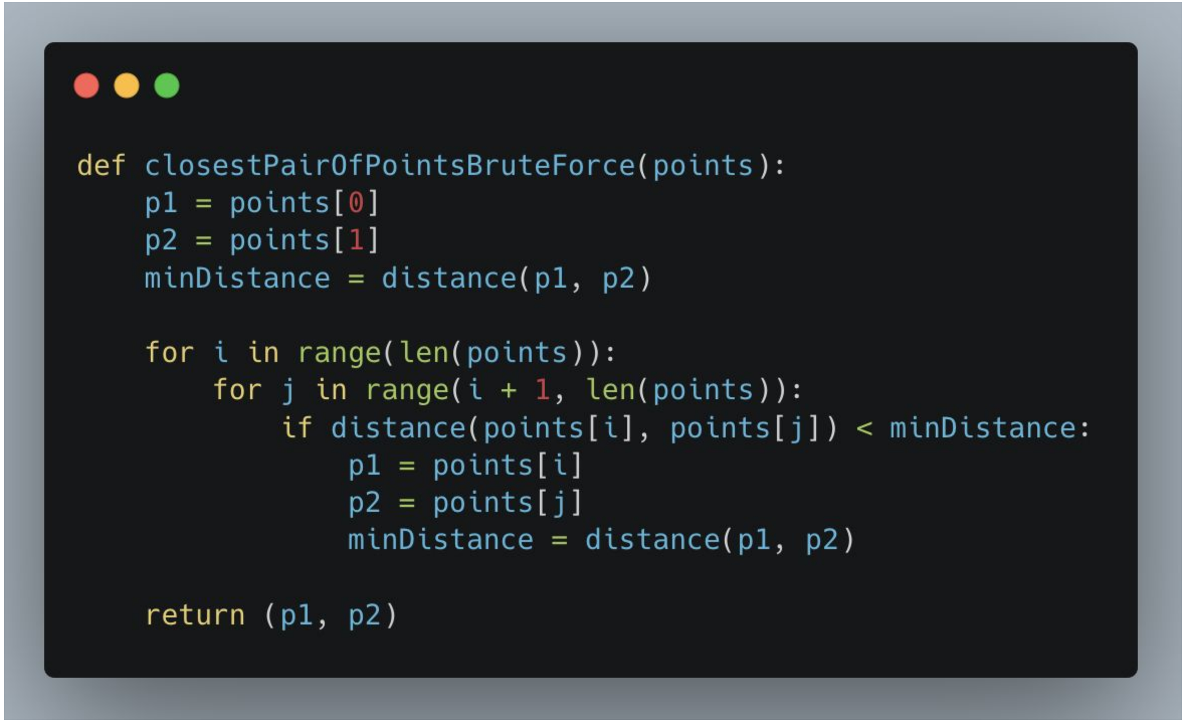
Problem statement

- Computational geometry problem
- Given n points in metric space, find a pair of points with the smallest distance between them
- Closest pair of points are shown in red
- Applications in computer graphics, computer vision, air traffic control, etc



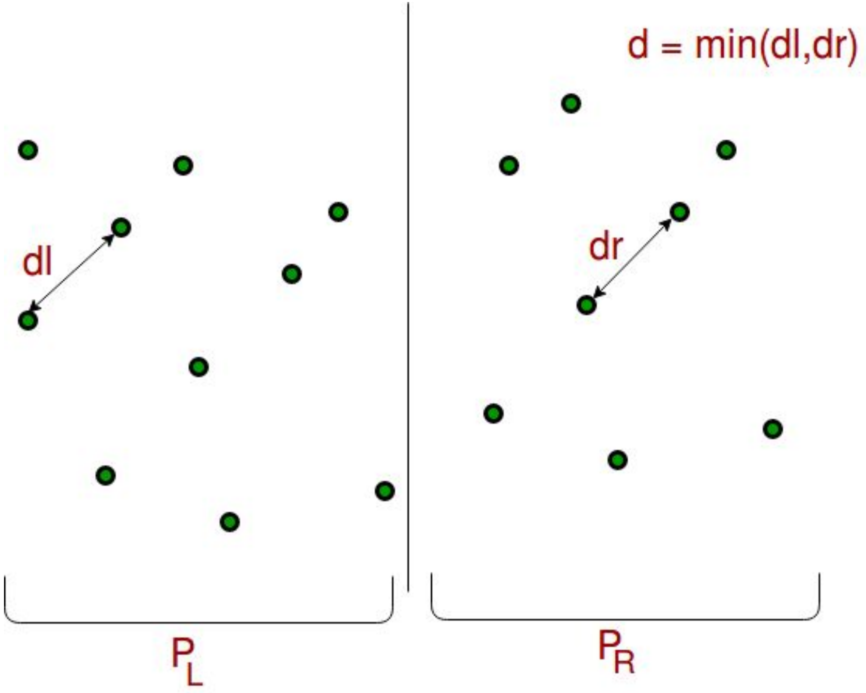
Wikipedia contributors, "Closest pair of points problem," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Closest_pair_of_points_problem&oldid=1136145228 (accessed March 30, 2023).

Brute force solution



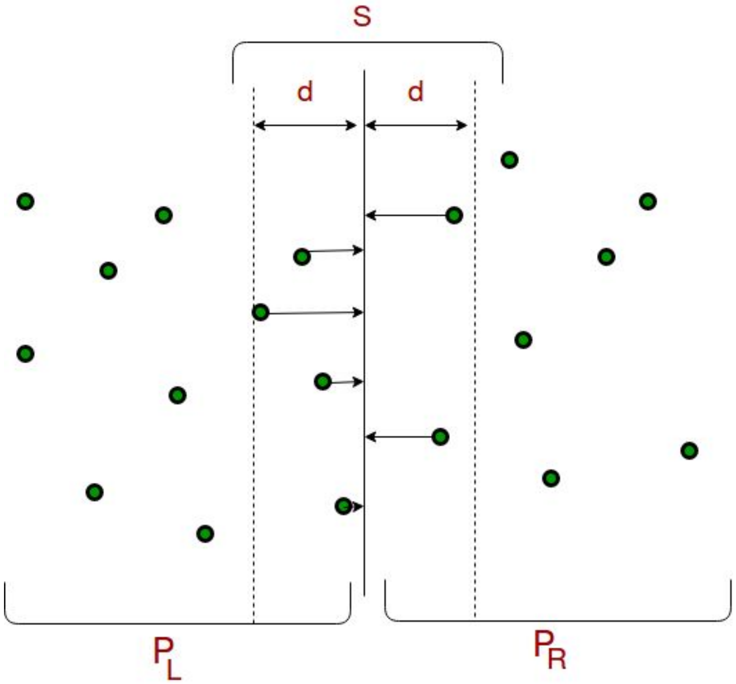
```python
def closestPairOfPointsBruteForce(points):
    p1 = points[0]
    p2 = points[1]
    minDistance = distance(p1, p2)

    for i in range(len(points)):
        for j in range(i + 1, len(points)):
            if distance(points[i], points[j]) < minDistance:
                p1 = points[i]
                p2 = points[j]
                minDistance = distance(p1, p2)

    return (p1, p2)
```

Divide and Conquer solution



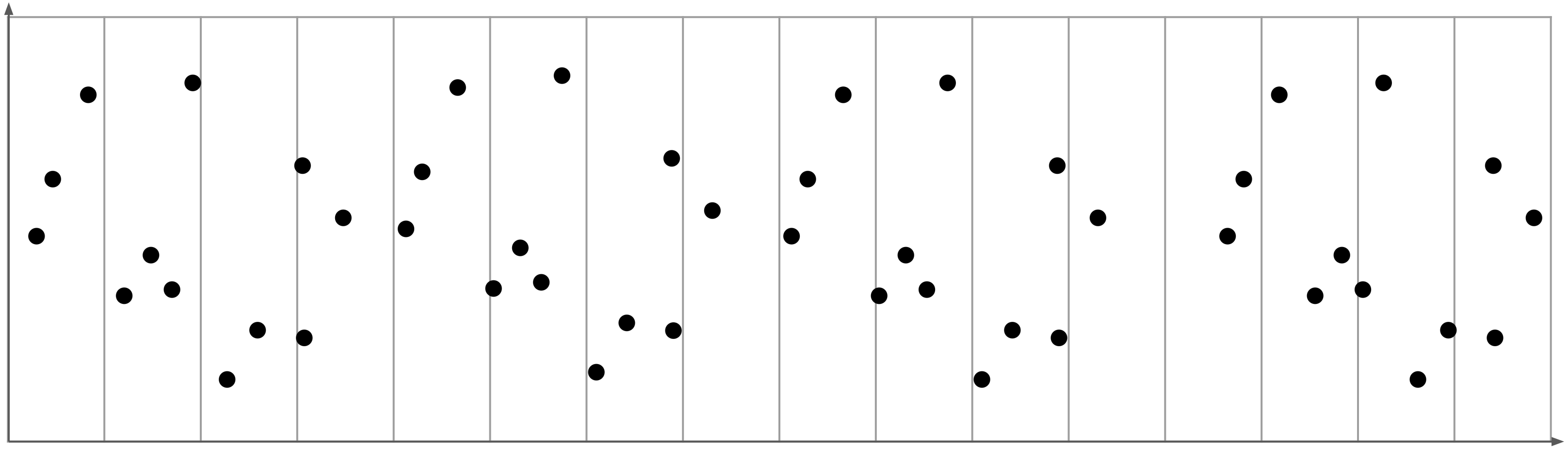
Closest pair of points using divide and conquer algorithm (2023) GeeksforGeeks. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm (Accessed: March 30, 2023).

Divide and Conquer solution

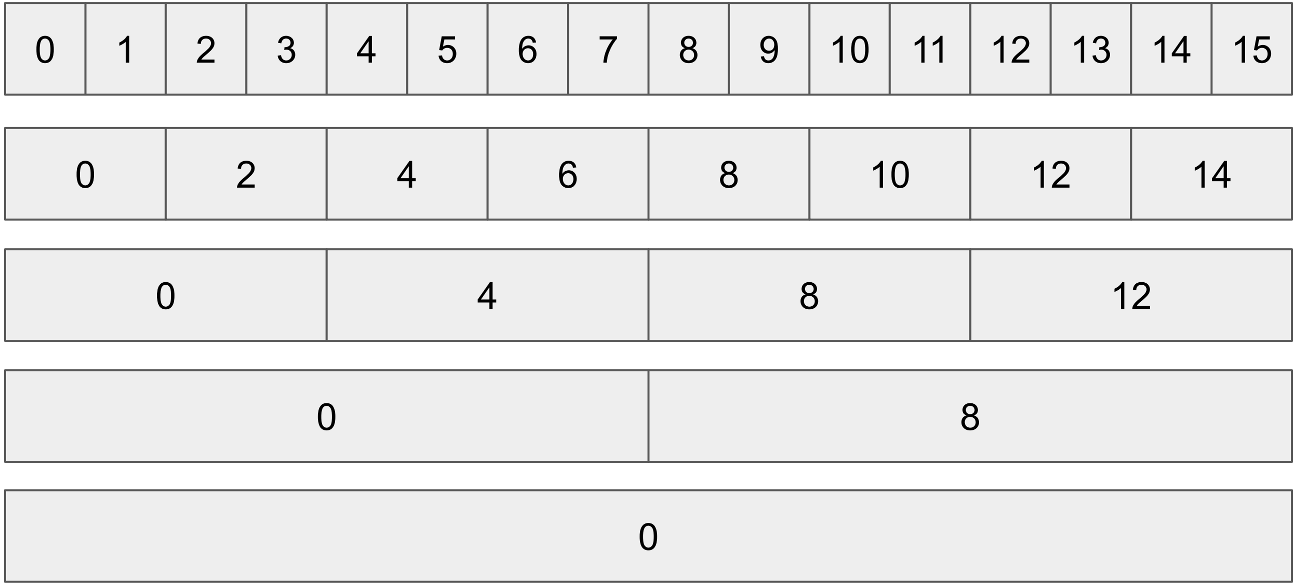


Closest pair of points using divide and conquer algorithm (2023) GeeksforGeeks. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm (Accessed: March 30, 2023).

Parallel implementation



Parallel implementation



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

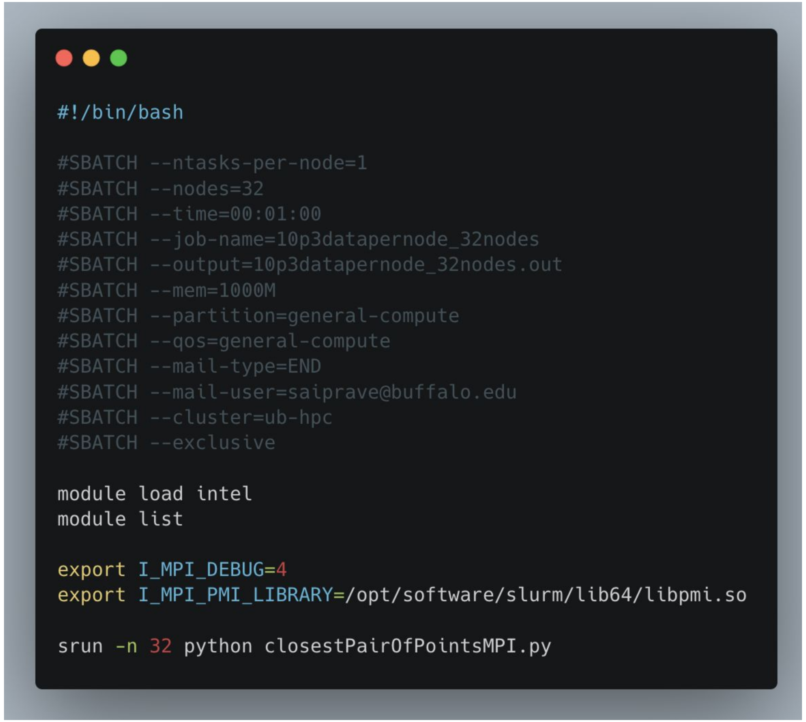
0 2 4 6 8 10 12 14

0 4 8 12

0 8

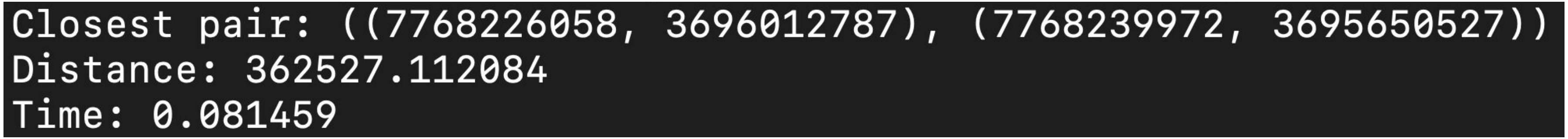
0

Slurm.sh



```bash
#!/bin/bash

#SBATCH --ntasks-per-node=1
#SBATCH --nodes=32
#SBATCH --time=00:01:00
#SBATCH --job-name=10p3datapernode_32nodes
#SBATCH --output=10p3datapernode_32nodes.out
#SBATCH --mem=1000M
#SBATCH --partition=general-compute
#SBATCH --qos=general-compute
#SBATCH --mail-type=END
#SBATCH --mail-user=saiprave@buffalo.edu
#SBATCH --cluster=ub-hpc
#SBATCH --exclusive

module load intel
module list

export I_MPI_DEBUG=4
export I_MPI_PMI_LIBRARY=/opt/software/slurm/lib64/libpmi.so

srun -n 32 python closestPairOfPointsMPI.py
```

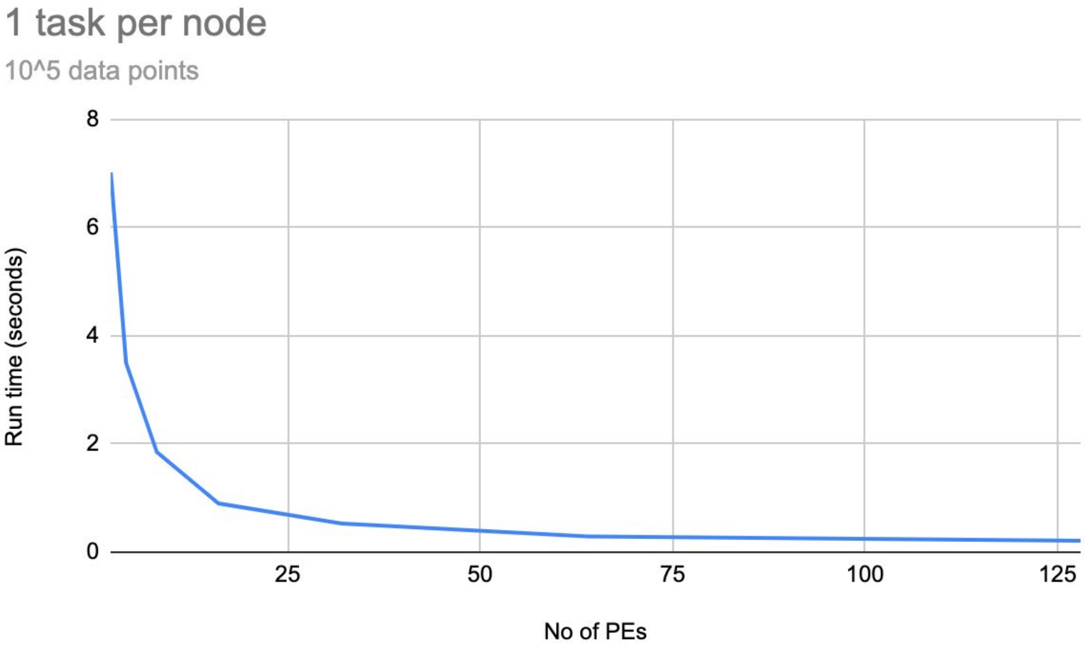
Sample output



```
Closest pair: ((7768226058, 3696012787), (7768239972, 3695650527))
Distance: 362527.112084
Time: 0.081459
```

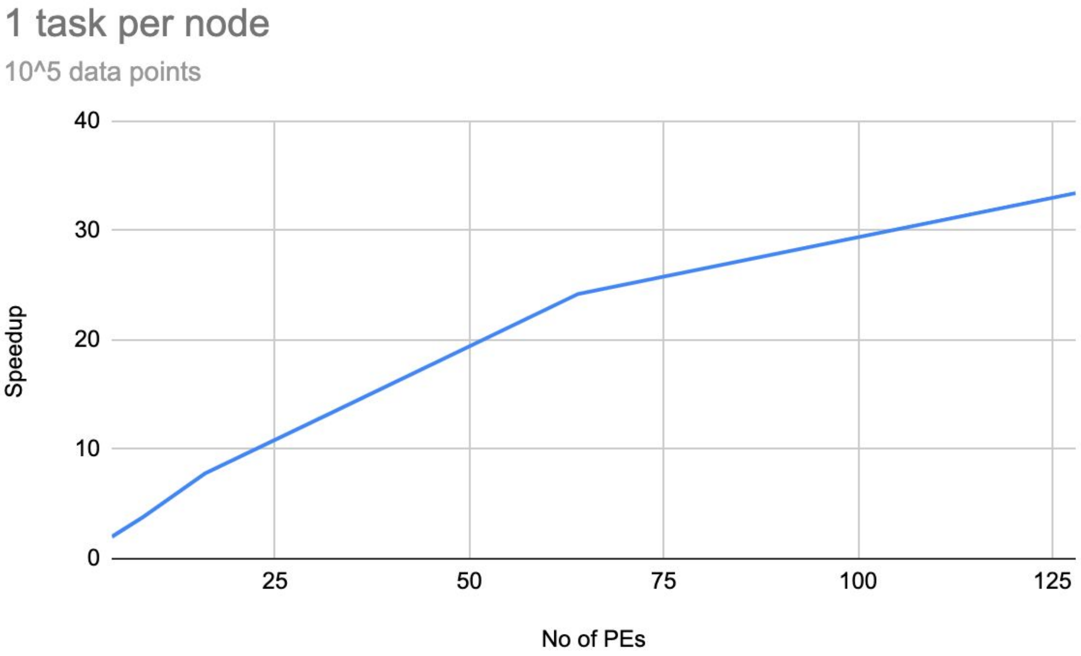
Results

No of nodes	Avg run time (sec)
2	7.02
4	3.5
8	1.85
16	0.9
32	0.53
64	0.29
128	0.21



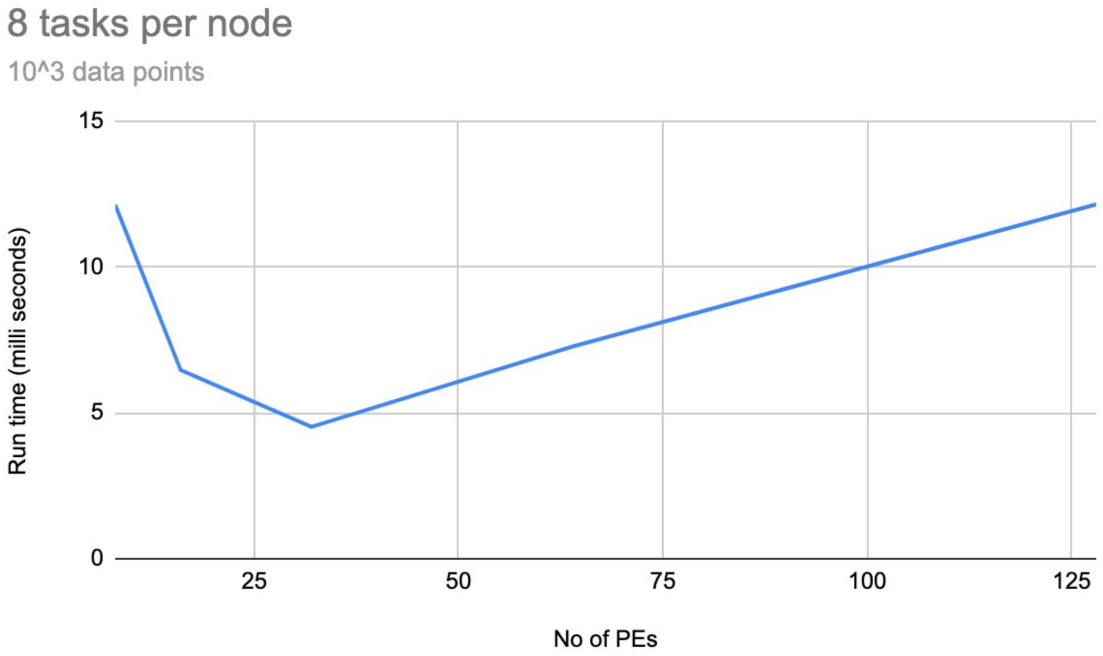
Results

No of nodes	Speedup
2	1
4	2.01
8	3.79
16	7.8
32	13.25
64	24.21
128	33.43



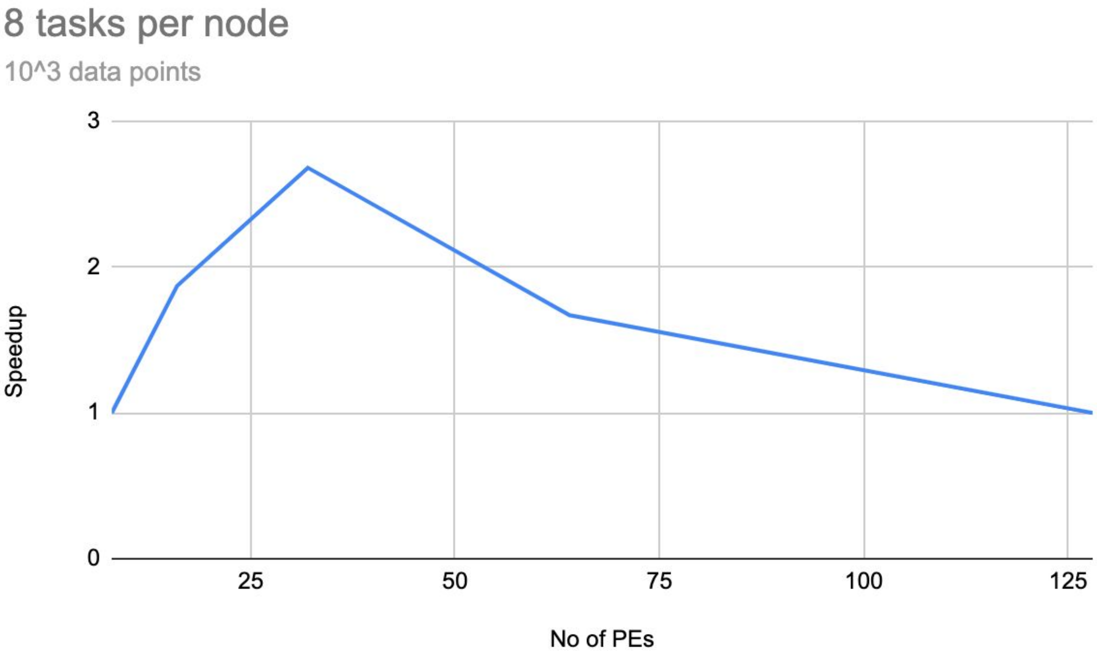
Results

No of PEs	Avg run time (ms)
8	12.13
16	6.47
32	4.52
64	7.28
128	12.15



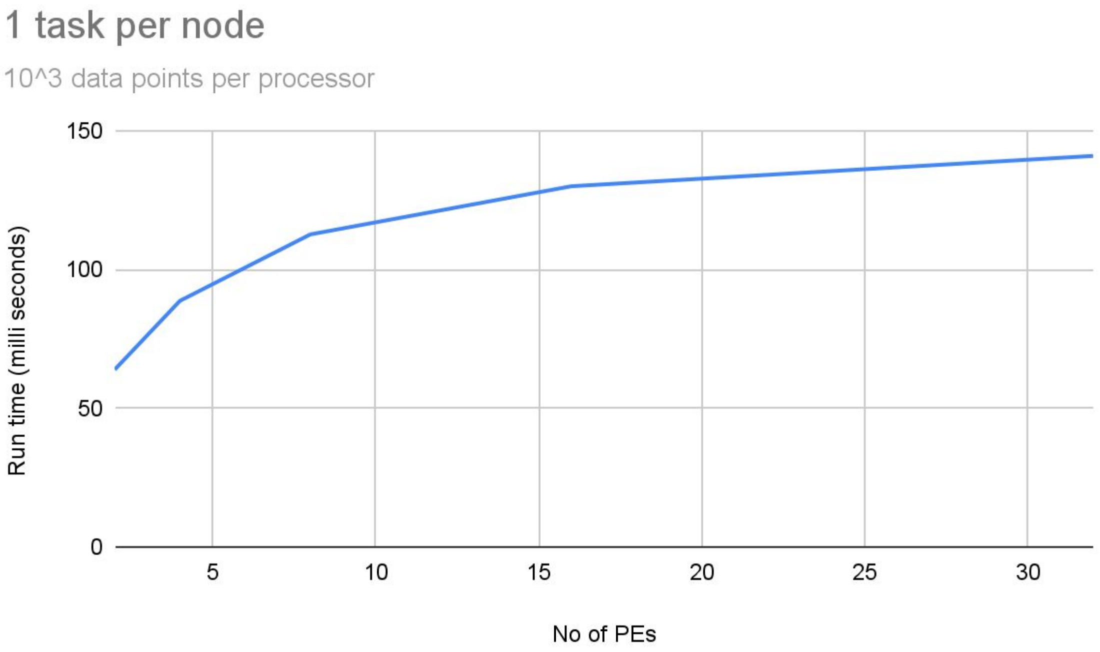
Results

No of PEs	Speedup
8	1
16	1.87
32	2.68
64	1.67
128	1



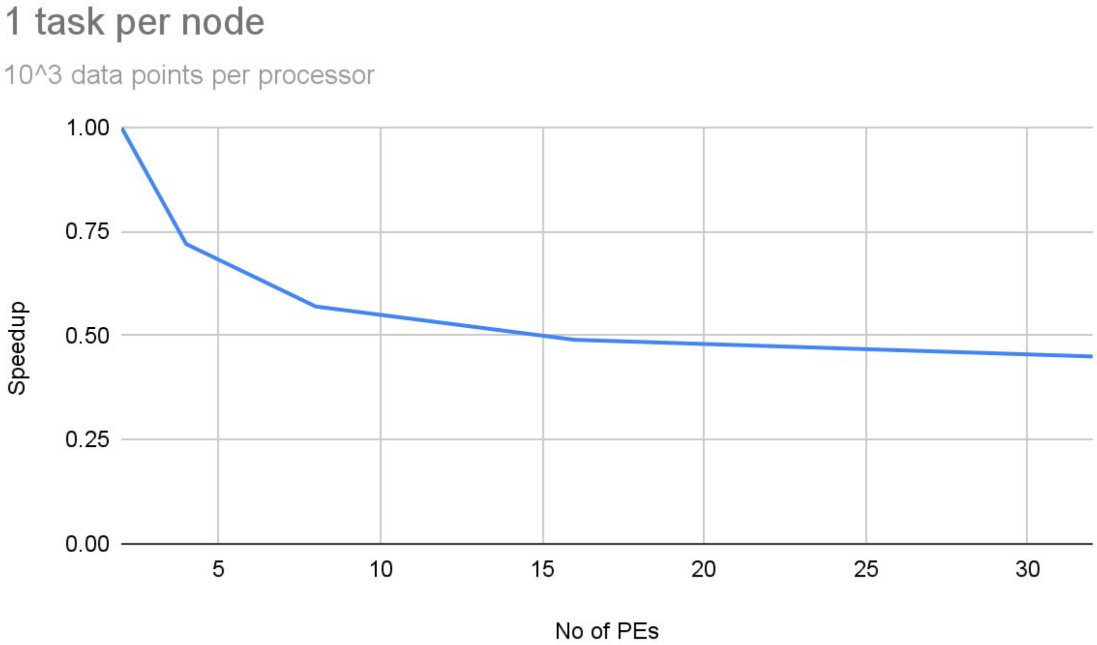
Results

No of PEs	Avg run time (ms)
2	63.92
4	88.71
8	112.66
16	130.01
32	140.95



Results

No of PEs	Speedup
2	1
4	0.72
8	0.57
16	0.49
32	0.45



References

- https://en.wikipedia.org/wiki/Closest_pair_of_points_problem
- https://mpi4py.readthedocs.io/en/stable/index.html
- https://docs.ccr.buffalo.edu/en/latest/
- https://ubccr.freshdesk.com/support/solutions/articles/13000026245-tutorials-and-training-documents

Thank You!