





nik



Trash

```
Open ▾  mpi_task.py Save  _ □ ×
```

```
10 parser = argparse.ArgumentParser()
11 parser.add_argument("points", help="display the total number of points", type=int)
12 args = parser.parse_args()
13 points = int(sys.argv[1])
14
15
16 def In_zone(x, y):
17     return (x*x+y*y)<1
18
19 def main():
20     comm = MPI.COMM_WORLD
21     rank = comm.Get_rank()
22     size = comm.size
23     points_to_calculate = points
24     in_zone = 0
25
26     if rank == 0:
27         arr = [[r.random(), r.random()] for i in range(points)]
28     else:
29         arr = None
30         points_to_calculate = int((points/size)*rank)
31
32     arr = comm.bcast(arr, root=0)
33     start = MPI.Wtime()
34
35     for i in range(int(points_to_calculate)):
36         in_zone += In_zone(arr[i][0], arr[i][1]);
37     pi = 4.0 * in_zone / points_to_calculate;
38     end = MPI.Wtime()
39
40
41     ans = dict()
42     if rank != 0:
```

Python ▾ Tab Width: 8 ▾ Ln 57, Col 42 ▾ INS



```

32 arr = comm.bcast(arr, root=0)
33 start = MPI.Wtime()
34
35 for i in range(int(points_to_calculate)):
36     in_zone += In_zone(arr[i][0],arr[i][1]);
37 pi = 4.0 * in_zone / points_to_calculate;
38 end = MPI.Wtime()
39
40
41 ans = dict()
42 if rank != 0:
43     data = [rank, points_to_calculate, end-start, pi, abs(PI-pi)]
44     comm.send(data, dest=0)
45
46 else:
47     ans[rank] = [points_to_calculate, end-start, pi, abs(PI-pi)]
48     for k in range(1, size):
49         val = comm.recv(source=k)
50         ans[val[0]] = val[1:]
51
52 if rank == 0:
53     for j in range(1, size):
54         print(f"Rank: {j}\tPoints_to_calculate: {ans[j][0]}\nTime: {ans[j][1]}\n\tpi: {ans[j][2]}\t Error: {ans[j][3]}\n\n" )
55         print(f"Rank: {0}\tPoints_to_calculate: {ans[0][0]}\nTime: {ans[0][1]}\n\tpi: {ans[0][2]}\t Error: {ans[0][3]}\n\n" )
56
57 #mpirun -np 2 python3 mpi_task.py 1000000
58
59
60
61 main()
62

```

Python ▾ Tab Width: 8 ▾ Ln 59, Col 1 ▾ INS

```
nik@ubuntu: ~  
/home/nik/catkin_ws/src/opt/ros/noetic/share  
nik@ubuntu:~$ mpirun -np 2 python3 mpi_task.py 1000000  
Rank: 1 Points_to_calculate: 500000  
Time: 0.100439416      pi: 3.138976      Error: 0.0026166535897931276  
  
Rank: 0 Points_to_calculate: 1000000  
Time: 0.189604167      pi: 3.138884      Error: 0.0027086535897931086  
  
nik@ubuntu:~$
```



```
mpi_task.py
32 arr = comm.bcast(arr, root=0)
33 start = MPI.Wtime()
34
35 for i in range(int(points_to_calculate)):
36     in_zone += In_zone(arr[i][0],arr[i][1]);
37 pi = 4.0 * in_zone / points_to_calculate;
38 end = MPI.Wtime()
39
40
41 ans = dict()
42 if rank != 0:
43     data = [rank, points_to_calculate, end-start, pi, abs(PI-pi)]
44     comm.send(data, dest=0)
45
46 else:
47     ans[rank] = [points_to_calculate, end-start, pi, abs(PI-pi)]
48     for k in range(1, size):
49         val = comm.recv(source=k)
50         ans[val[0]] = val[1:]
51
52 if rank == 0:
53     for j in range(1, size):
54         print(f"Rank: {j}\tPoints_to_calculate: {ans[j][0]}\nTime: {ans[j][1]}\n\tpi: {ans[j][2]}\t Error: {ans[j][3]}\n\n")
55     print(f"Rank: {0}\tPoints_to_calculate: {ans[0][0]}\nTime: {ans[0][1]}\n\tpi: {ans[0][2]}\t Error: {ans[0][3]}\n\n")
56
57 #mpirun -np 2 python3 mpi_task.py 1000000
58
59
60
61 main()
62
```

Python Tab Width: 8 Ln 59, Col 1 INS

```
nik@ubuntu: ~
nik@ubuntu:~$ mpirun -np 4 python3 mpi_task.py 1000000
Rank: 1 Points_to_calculate: 250000
Time: 0.115087919 pi: 3.144656 Error: 0.0030633464102067798

Rank: 2 Points_to_calculate: 500000
Time: 0.261566251 pi: 3.142688 Error: 0.001095346410207032

Rank: 3 Points_to_calculate: 750000
Time: 0.354212111 pi: 3.1408106666666667 Error: 0.0007819869231262544

Rank: 0 Points_to_calculate: 1000000
Time: 0.384183704 pi: 3.1395 Error: 0.0020926535897931586

nik@ubuntu:~$
```




```
mpi_task.py
32 arr = comm.bcast(arr, root=0)
33 start = MPI.Wtime()
34
35 for i in range(int(points_to_calculate)):
36     in_zone += In_zone(arr[i][0], arr[i][1]);
37 pi = 4.0 * in_zone / points_to_calculate;
38 end = MPI.Wtime()
39
40
41 ans = dict()
42 if rank != 0:
43     data = [rank, points_to_calculate, end-start, pi, abs(PI-pi)]
44     comm.send(data, dest=0)
45
46 else:
47     ans[rank] = [points_to_calculate, end-start, pi, abs(PI-pi)]
48     for k in range(1, size):
49         val = comm.recv(source=k)
50         ans[val[0]] = val[1:]
51
52 if rank == 0:
53     for j in range(1, size):
54         print(f"Rank: {j}\tPoints_to_calculate: {ans[j][0]}\nTime: {ans[j][1]}\n\tpi: {ans[j][2]}\tError: {ans[j][3]}\n\n")
55     print(f"Rank: {0}\tPoints_to_calculate: {ans[0][0]}\nTime: {ans[0][1]}\n\tpi: {ans[0][2]}\tError: {ans[0][3]}\n\n")
56
57 #mpirun -np 2 python3 mpi_task.py 1000000
58
59
60
61 main()
62
```

Python Tab Width: 8 Ln 59, Col 1 INS

```
nik@ubuntu: ~
Rank: 1 Points_to_calculate: 500000
Time: 0.100439416 pi: 3.138976 Error: 0.0026166535897931276

Rank: 0 Points_to_calculate: 1000000
Time: 0.189604167 pi: 3.138884 Error: 0.0027086535897931086

nik@ubuntu:~$ mpirun -np 6 python3 mpi_task.py 1000000
Rank: 1 Points_to_calculate: 166666
Time: 0.074533143 pi: 3.1358285433141733 Error: 0.005764110275619849

Rank: 2 Points_to_calculate: 333333
Time: 0.137043853 pi: 3.141987141987142 Error: 0.00039448839734879115

Rank: 3 Points_to_calculate: 500000
Time: 0.44293697 pi: 3.14276 Error: 0.001167346410206882

Rank: 4 Points_to_calculate: 666666
Time: 0.31219789 pi: 3.142899142899143 Error: 0.0013064893093499563

Rank: 5 Points_to_calculate: 833333
Time: 0.286130057 pi: 3.142220456888183 Error: 0.0006278032983897397

Rank: 0 Points_to_calculate: 1000000
Time: 0.224678688 pi: 3.143568 Error: 0.001975346410207024

nik@ubuntu:~$
```