实验3

实验环境

macOS 10.14，mpirun (Open MPI) 4.0.3，g++-9 (Homebrew GCC 9.3.0_1) 9.3.0

由于电脑线程不够（仅有2核4线程），在使用mpi时使用了`oversubscribe`，所以有部分无法运行。

算法设计与分析

将所有物体分组交由各个线程模拟，来实现并行模拟。每个周期结束都将信息进行广播。

核心代码

```
for (int i = 0; i < steps; i++) {
        for (int j = N / size * rank; j < N / size * rank + N / size; j++) {
            force(x, y, j, fx, fy);
            velocities(vx, vy, fx, fy, j);
            positions(x, y, vx, vy, j);
        }
        if (size > 1) {
            for (int j = 0; j < size; j++) {
                MPI_Bcast(&x[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
                MPI_Bcast(&y[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
                MPI_Bcast(&vx[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
                MPI_Bcast(&vy[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
                MPI_Bcast(&fx[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
                MPI_Bcast(&fy[N / size * j], N / size, MPI_LONG_DOUBLE, j,
MPI_COMM_WORLD);
            }
        }
    }
```

实验结果

规模/进程数	1	2	4	8
64	0.113525	0.072841/1.55853	0.108974/1.04176	0.154027/0.737046
256	1.932948	0.981457/1.96947	0.905900/2.13373	1.499130/1.28938

运行时间(sec)/加速比

分析与总结

在模拟中，并行有明显的效果，尤其是对于较多的物体。物体数越多，并行加速比越大，且最佳的进程数越高。我还测试了N=1024, 2048时的情况，都验证了这一结论，最大的限制应该是我的电脑只有双核四线程，性能受限。