实验2

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

将所有车辆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 < cycle; i++) {
        for (int j = rank * car / size; j < rank * car / size + car / size; j++)
{
            if (j == car - 1 && v[j] < v_max) {
                v[j] ++;
            }
            else if (j != car - 1) {
                if (v[j] < v_max) {
                    v[j] ++;
                }
                distance = position[j + 1] - position[j];
                if (distance <= v[j]) {
                    v[j] = distance - 1;
                }
            }
            randNum = rand() / (double)RAND_MAX;
            if (randNum < p) {
                if (v[j] > 0) {
                    v[j]--;
                }
            }
            if (j == car - 1)
            {
                position[j] += v[j];
            }
            else
            {
                if (position[j] + v[j] < position[j + 1])
                {
                    position[j] += v[j];
```

```
                }
                else
                    position[j] = position[j + 1] - 1;
            }
        }
        if (size > 1) {
            for (int j = 0; j < size; j++) {
                MPI_Bcast(&position[j * car / size], car / size, MPI_LONG, j,
MPI_COMM_WORLD);
                MPI_Bcast(&v[j * car / size], car / size, MPI_INT, j,
MPI_COMM_WORLD);
            }
        }
    }
```

实验结果

规模/进程数	1	2	4	8
100,000	6.184932	4.226426/1.4634	9.874518/0.626353	34.964622/0.176891
500,000	6.947340	4.314070/1.61039	10.774395/0.644801	28.039077/0.247773
1,000,000	8.473499	5.343728/1.58569	9.676018/0.875722	19.684866/0.430458

运行时间(sec)/加速比

分析与总结

观察实验结果可以发现，使用两个线程得到的结果最好，而且呈现出模拟周期越少，车辆数量越多，并行效果越好的趋势。分析结果和代码，可以看出由于每个周期结束都要进行广播，也就是进程间通信，拖慢了整体运行的速度。使用两个线程可以达到平衡的最佳点。也可以进行合理推断，如果周期进一步减少，或者车辆规模进一步增加，使用更多线程就会有更好的加速比。