

实验 2

PB16001715 陈思源

2020 年 7 月 14 日

1 实验题目

利用 MPI 进行蒙特卡洛模拟。

在道路交通规划上，需要对单条道路的拥堵情况进行估计。因为仅考虑单条车道，所以不存在超车。假设共有 n 辆车，分别编号 $0, 1, \dots, n-1$ ，每辆车占据一个单位的空间。初始状态如下， n 辆车首尾相连，速度都是 0。每个时间周期里每个车辆的运动满足以下规则：

1. 假设当前周期开始时，速度是 v 。
2. 和前一辆车的距离为 d （前一辆车车尾到这辆车车头的距离，对于第 0 号车， $d = \infty$ ），若 $d > v$ ，它的速度会提高到 $v+1$ 。最高限速 v_{max} 。若 $d \leq v$ ，那么它的速度会降低到 d 。
3. 前两条完成后，司机还会以概率 p 随机减速 1 个单位。速度不会为负值。
4. 基于以上几点，车辆向前移动 v （这里的 v 已经被更新）个单位。

2 实验环境

运行在服务器节点上，操作系统内核 Linux 3.10.0-862.el7.x86_64，使用 gcc 编译器，版本 4.8.5。处理器为 Intel 至强 E5-2620，基准频率 2.00GHz。

3 算法设计与分析

输入：模拟的车辆数 car_num 和周期数 $periods$

输出：模拟结束后各车辆的位置

资源： p 个进程

解题思路

将所有车辆均匀划分为块，分配给每个进程。每个进程在一个周期结束后将最后一辆车的位置发送给下一个进程，为下个进程的第一辆车的位置更新提供信息。

实现步骤

1. 将编号为 0 到 $car_num - 1$ 的车辆均匀划分，将每个大小为 $\frac{n}{p}$ 的块分配给一个进程。
2. 编号为 $i (i \neq p - 1)$ 的进程向编号为 $i + 1$ 的进程发送最后一辆车的位置。
3. 编号为 $i (i \neq 0)$ 的进程接收编号为 $i - 1$ 的进程最后一辆车的位置，依次更新每辆车的速度和位置。编号为 0 的进程的第 0 号车 $d = \text{无穷大}$ 。
4. 重复 2、3 $periods$ 个周期，然后主进程收集各个进程的车辆位置信息并输出。

4 核心代码

```
1 for (i = 0; i < periods; i++)
2 {
3     // 向后面的块发送最后一辆车的位置
4     if (rank != size - 1)
5     {
6         MPI_Send(dist+car_local_num-1, 1, MPI_INT, rank+1, i,
7                 MPI_COMM_WORLD);
8     }
9     // 接受前一块最后一辆车的位置并计算距离
10    if (rank == 0)
11    {
12        d = -car_num;    // 代表无穷远
13    }
14    else
15    {
16        MPI_Recv(&d, 1, MPI_INT, rank-1, i, MPI_COMM_WORLD, &status);
17        d = d - dist[0] - 1;
18    }
19    // 更新速度
20    v[0] = next_v(v[0], d, car_num);
21    for (j = 1; j < car_local_num; j++)
22    {
23        d = dist[j-1] - dist[j] - 1;
24        v[j] = next_v(v[j], d, car_num);
25    }
```

```

25 // 更新位置
26 dist[0] += v[0];
27 for (j = 1; j < car_local_num; j++)
28 {
29     dist[j] += v[j];
30 }
31 }

```

Listing 1: 蒙特卡洛模拟核心代码

5 实验结果

对每个规模和进程数运行三次并取平均，得到结果如下。

表 1: 蒙特卡洛模拟的实验结果

(a) 运行时间 (s)

进程数 车辆, 周期	1	2	4	8
(100000, 2000)	3.942322	2.035079	1.126165333	0.681423
(500000, 500)	4.840742333	2.523566333	1.377657667	0.821606
(1000000, 300)	5.775139667	3.004904333	1.613401333	0.924265333

(b) 加速比

进程数 车辆, 周期	1	2	4	8
(100000, 2000)	1	1.9372	3.0066	5.7854
(500000, 500)	1	1.9182	3.5137	5.8918
(1000000, 300)	1	1.9219	3.5795	6.2484

6 分析与总结

本实验的蒙特卡洛模拟程序在多线程上取得了较好的加速比，且规模越大，多线程加速效果越好。程序的进程间通信需求很少，只需链式传递一个整型变量，多个进程按流水线方式前进，因此可以取得较好的加速效果。当进程数相同时，规模越大，通信占比越小，加速比越高。