

# GPU 编程实验 2--图采样

学号	专业班级	姓名
3200100574	图灵2001	宋佳铮

## 1. Project Introduction

本次实验是利用GPU并行实现图神经网络随机采样(sec1)，并优化统计以40个节点为一个bin的总被采样次数的核函数。本次实验运行在RTX8000上。

## 2. Technical Details

### sec1:

主要实现随机采样函数的并行，最关键的是搞清楚CSR存储格式：

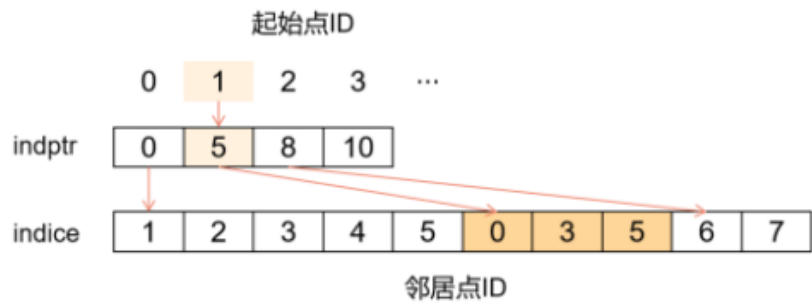


图1：CSR存储格式

即根据初始节点的ID在下标指针数组（indptr）中找到邻居节点存储地址的起始位置，并以下一个节点的邻居节点存储地址的起始位置为邻居节点的结束位置，获得节点的所有邻居节点ID。

随后就是最基础的多线程跨网格并行代码框架，需要注意的是随机采样时生成的随机数是根据threadIdx.x来的，要么修改随机采样的函数使得其在一起起始点的多次采样结果不同，要么修改线程下标定义：根据每次采样开线程，而不是每个节点开线程。

同时需要注意的是代码复用性，在编号采样节点存储地址时不要采用硬编码，采用函数输入。

### sec2:

主要实现对于超大数组（第一跳目标节点）的信息进行并行统计的函数的优化加速。

原始的函数已经实现了开满线程并行读取目标节点数组，但由于写入的统计数组是一个全局数组，并且需要保证写入操作的原子性，因此在遇到同一个位置需要写入时就退化成了串行。

因此第一个考虑优化的地方是使用中间临时存储数组存储块内统计结果，最后将所有块内的统计数组累加获得最终全局结果。

第二个想法是将输入的d\_dst\_ids数组在对应块需要索引的数据读入块内共享内存，但是仔细思考发现就算读入每个block的共享内存也至少需要block内的每个线程读一次，由于读取的顺序性以及一次读取的不是一个int，这反而会影响到预取缓存的命中率，因此这个优化其实没有什么意义甚至是负优化。基于相同的原因交错寻址没能实现优化。

### 3. Experiment Results

sec1

```
1    0 0 196364
2    0 1 225676
3    0 2 182894
4    0 3 27588
5    1 0 125582
6    1 1 165193
7    1 2 22136
8    1 3 220082
9    2 0 5440
10   2 1 119391
11   2 2 5440
12   2 3 119391
13   3 0 40339
14   3 1 6587
15   3 2 40339
16   3 3 40339
17   4 0 187588
18   4 1 219210
19   4 2 149978
20   4 3 214036
```

图2: 前20个随机采样点ID

```
396   98 3 182956
397   99 0 139531
398   99 1 154349
399   99 2 91499
400   99 3 220140
401   time cost: 0.367392 ms
```

图3: 采样 GPU 核函数执行时间

sec2

```
96    95 38486
97    96 32664
98    97 45980
99    98 33549
100   99 40747
101   time cost: 3.52442 ms
```

图4: base版本核函数执行时间

```

96    95 38486
97    96 32664
98    97 45980
99    98 33549
100   99 40747
101   time cost: 2.07219 ms

```

图5：优化后核函数执行时间

采样方式不同（每个线程采样一次还是采样多次），最终时间也不一致，上面是每个线程采样多次的结果。

```

96    95 38580
97    96 32363
98    97 45795
99    98 33342
100   99 41094
101   time cost: 4.76867 ms

```

```

96    95 38580
97    96 32363
98    97 45795
99    98 33342
100   99 41094
101   time cost: 2.16422 ms

```

图6-7：base版本核函数执行时间和优化后核函数执行时间

一个线程采样一次的结果如上。

可以直接查看提交结果，其中sec2\_base为一个线程多次采样未优化版本的输出，sec2\_opt为优化后输出。sec2\_sbase为一个线程采样1次未优化版本的输出，sec2\_sopt为其优化后输出。采用一个线程多次采样的采样方式最终前100个节点采样结果如下

1	0	34308
2	1	27024
3	2	43023
4	3	54030
5	4	39644
6	5	31130
7	6	57252
8	7	43606
9	8	29800
10	9	42712
11	10	34803
12	11	34162
13	12	56015
14	13	43090
15	14	37084
16	15	46958
17	16	41483
18	17	33672
19	18	93376
20	19	32758
21	20	61525

22	21	30441
23	22	37383
24	23	33963
25	24	49831
26	25	31357
27	26	42964
28	27	46744
29	28	45144
30	29	31728
31	30	48798
32	31	46060
33	32	59779
34	33	32565
35	34	46076
36	35	52296
37	36	30045
38	37	52080
39	38	31551
40	39	43638
41	40	56339
42	41	33463
43	42	43163
44	43	39293
45	44	36882
46	45	47629
47	46	58128
48	47	37379
49	48	37314
50	49	42817
51	50	30845
52	51	55083
53	52	28618
54	53	37378
55	54	28192
56	55	40838
57	56	56647
58	57	41019
59	58	38264
60	59	50161
61	60	38947
62	61	51770
63	62	34610
64	63	48204
65	64	40509
66	65	42207
67	66	34152
68	67	38950
69	68	45043
70	69	69752
71	70	35815
72	71	56745
73	72	51247
74	73	36563
75	74	36621
76	75	43908
77	76	36236
78	77	34291
79	78	37844

80	79	40965
81	80	31500
82	81	44517
83	82	29971
84	83	42459
85	84	31426
86	85	33227
87	86	49868
88	87	42980
89	88	32738
90	89	52243
91	90	42132
92	91	47500
93	92	50947
94	93	37154
95	94	41483
96	95	38486
97	96	32664
98	97	45980
99	98	33549
100	99	40747