

第五次的作业

1. 解读课上并行scan算法，给出其原理，证明正确性；
2. 对于碰撞检测场景，如果每个线程检测出来的碰撞结果个数完全随机，如何实现结果收集（连续保存在一个大小为M的数组，M为总的碰撞个数）？

1.1 并行scan算法解读

scan 算法的计算伪代码为

```
for j:=1 to log2n do
  for all k in parallel do
    if k >= 2^j then
      x[k]:=x[k-2^(j-1)] + x[k]
    fi
  od
od
```

其中 j 表示迭代计算的次数，每次计算时检查当前数据的索引值 k 与 $2^{(j-1)}$ 进行比较，若有 $k \geq 2^{(j-1)}$ ，则当前下标为 k 的数据与 $k - 2^{(j-1)}$ 相加作为 $j + 1$ 轮下标为 k 处的数据，若条件不成立， $j + 1$ 轮下标为 k 处的数据与 j 轮数据一致。直至 j 的值与 $\lfloor \log_2 n \rfloor$ 的值一致，即 $2^{j-1} < n \leq 2^j$ ，运算结束。最终的值与串行计算结果一致， k 处结果为 $\sum_{i=0}^k x_i$ 。

上述计算实现了层序分离，使得每一轮的计算能够利用到 $n - 2^j$ 数量的线程，相较串行办法，总的加法计算复杂度为 $O(n \log n)$ ，每一步的计算复杂度为 $O(\log n)$ 。

1.2 Hillis Steele Algorithm 证明

根据Hillis Steele算法原理，有如下关系成立：对于一个 k 维的并行计算数据，记 x_n 为第 n 个数， j 表示第 j 次运算，那么有

$$\begin{cases} x_n^{[j]} = x_{n-2^{j-1}}^{[j-1]} + x_n^{[j-1]} & , n > 2^{j-1}, j \in [1, \log_2 k] \\ x_n^{[j]} = x_n^{[j-1]} & , n \leq 2^{j-1}, j \in [1, \log_2 k] \end{cases} \quad (1)$$

不妨记区间 $[a, b]$ 之间的元素和为

$$\sum_{(a)}^{(b)} = \sum_{i=a}^b x_i, i \in [a, b]$$

那么有如下等式必然成立

$$\sum_{(a)}^{(b)} + \sum_{(b+1)}^{(c)} = \sum_{(a)}^{(c)}$$

式 (1) 中的上式可以表示为如下形式

$$x_n^{[j_f]} = \sum_{(0)}^{(n)} = \sum_{i=0}^1 x_{n-i*2^{j-1}}^{[j-1]}$$

证明过程

根据式 (1)，推断向前 m 步展开的结果应为

$$x_n^{[j_f]} = \sum_{(0)}^{(n)} = \sum_{i=0}^{2^m-1} x_{n-i*2^{j-m}}^{[j-m]} \quad (2)$$

对式 (2) 代入 式 (1) 的上式，再向前展开一步，即向前展开到 $m+1$ 步，如下所示

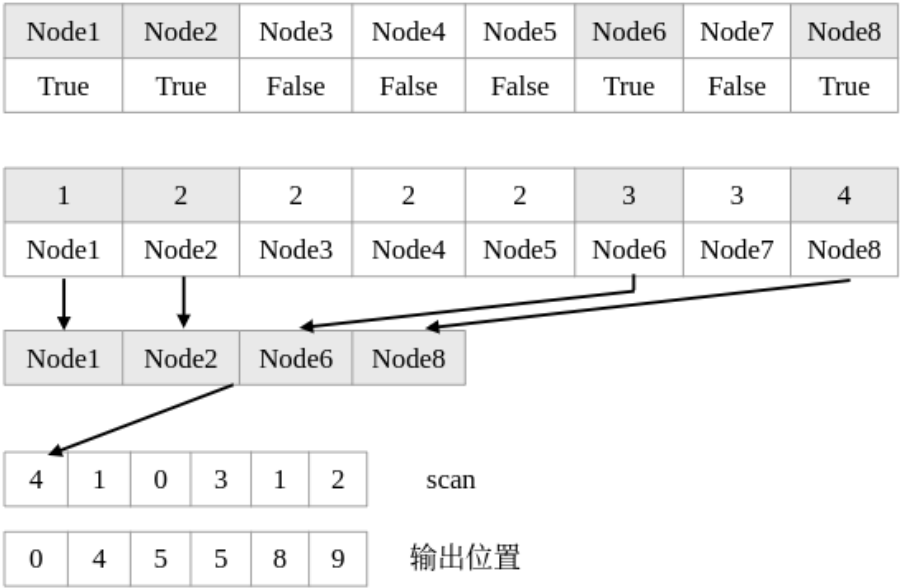
$$\begin{aligned} \sum_{i=0}^{2^m-1} x_{n-i*2^{j-m}}^{[j-m]} &= \sum_{i=0}^{2^m-1} x_{n-i*2^{j-m}-2^{j-m-1}}^{[j-m-1]} + x_{n-i*2^{j-m}}^{[j-m-1]} \\ &= \sum_{i=0}^{2^m-1} x_{n-(2i+1)*2^{j-m-1}}^{[j-m-1]} + x_{n-(2i)*2^{j-m-1}}^{[j-m-1]} \\ &= x_n^{[j-m-1]} + x_{n-2^{j-m-1}}^{[j-m-1]} + x_{n-2*2^{j-m-1}}^{[j-m-1]} + x_{n-3*2^{j-m-1}}^{[j-m-1]} + \dots + x_{n-(2^{m+1}-2)*2^{j-m-1}}^{[j-m-1]} + x_{n-(2^{m+1}-1)*2^{j-m-1}}^{[j-m-1]} \\ &= \sum_{i=0}^{2^{m+1}-1} x_{n-i*2^{j-m-1}}^{[j-m-1]} \end{aligned} \quad (3)$$

通过数学归纳法代入式 (3) 递推公式，可以证明式 (2) 为该公式的通项，当 $m=j$ 时有

$$x_n^{[j_f]} = \sum_{(0)}^{(n)} = \sum_{i=0}^{2^j-1} x_{n-i*2^0}^{[0]} = x_n^{[0]} + x_{n-1}^{[0]} + \dots + x_1^{[0]}, j \in MAX(1, \lfloor \log_2 k \rfloor) \quad (4)$$

证毕。

2. 碰撞检测结果存储



每个线程都去做碰撞检测，碰撞的节点记录碰撞结果 (True or False)，每个线程输出时，需要统计每个线程检测到的碰撞数量。那么对每个线程得到的碰撞数量进行 scan，则可以得到每个线程在输出数组的写入碰撞结果的起始位置。最终有M个碰撞点被检测到，最后的"carry"值也是M。

但对于老师课上讲的，这个办法能够避免开辟大量的固定大小的临时数组我有些疑问。

疑问： 用scan去判断存储位置之前，这些碰撞结果存储在哪里（应该也是个临时的空间）？scan算法调用的前提也是碰撞结果都已知，并且碰撞结果数量已知，那结果肯定存在一个地方，如何实现课上说的不需要为每一个线程预先开辟固定数组以保存碰撞结果的目标。