

并行与分布式计算导论大作业报告

程芷怡 元培学院 1800017781

2020 年 6 月 18 日

1 对问题的理解

本问题是为了模拟人脑而设计的。论文使用了Hines算法进行计算。串行算法利用了Hines矩阵的稀疏特性，将微分方程化简之后记录下每个矩阵中点的父节点，以及对应方程的系数，先后向再前向迭代计算。Hines矩阵稀疏化之后的结果形成了一棵树的结构。算法实际上相当于从树的叶节点开始向前迭代计算，再从树的根节点向后迭代计算，将树遍历了两次。

2 实现思路

观察得知，这棵树的节点可分为三类，一类是叶节点，一类是只有一个子节点的节点，一类是有多个子节点的节点。注意到，树的不同分支在计算上是相互独立的，因此可以将树的不同分支并行化。具体的思路是，在后向迭代的过程中，将多个子节点的节点当做入口。首先从叶节点开始进行多线程并行并后向迭代。当迭代到“入口”节点时返回主函数（Hines函数），并将这些入口节点当做下一次的“叶节点”进行递归计算。前向迭代的部分较为简单，只要在树分叉的地方进行多线程的递归计算即可（每个分支分配给一个线程进行计算）。在记录每个节点的儿子时，利用了两种方法。一种是数组记录（静态分配空间，观察发现由于是稀疏矩阵，猜想每个节点的儿子不会超过10个，并在试验中得到了证实）。另一种是链表记录（但是在递归计算时的主循环需要由一个线程进行，猜想可能会对性能造成影响）。两种方法的结果均在下表中有记录。

3 测试结果与分析

每个case做了100次试验取平均值： 观察可知，前6个case由于数据集

case	串行算法时间(ms)	并行算法1时间(ms)	并行算法2时间(ms)
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0.15625
6	0	0	0
7	3.28125	16.09375	0.15625
8	4.84375	15.3125	15.3125
9	17.34375	47.1875	14.0625
10	27.5	60.15625	36.5625
11	82.65625	169.84375	147.8125
12	166.25	361.25	320.15625

非常小，无法评测串行和并行算法的性能。从第7个case开始可以发现，采用数组存储的性能很不稳定，时好时坏，猜想可能和关键区的多少有关系。而采用链表的并行算法由于链表的遍历必须顺序执行，增加了额外的时间开销，并行性能并不是很好，加速比在0.2到0.4之间浮动。

4 可改进的地方和Future work

经过试验发现，这两个并行算法的可扩展性均一般，加速比很小。在调试代码的时候发现在后向遍历时由于需要添加barrier和保证每个“入口”节点的可进入性（即其子节点必须全部算完才能开始算）。猜想认为可以改进的地方是后向遍历，可以寻找更好的保证同步的算法。但是目前还没有更好的方案。