面向结构化网格的并行稀疏三角求解器SpTRSV

陈道琨、刘芳芳、杨超
联系人：刘芳芳 fangfang@iscas.ac.cn

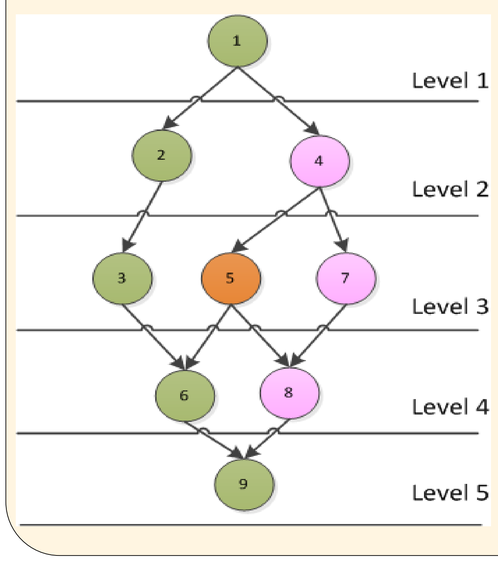
1 背景介绍-SpTRSV

- 稀疏三角求解器SpTRSV即求解Lx = b 或 Ux = b（L/U为稀疏三角矩阵），应用广泛
- 其计算过程有很强的依赖关系，并行难度大
- 其为访存密集型问题，在目前的处理器架构上优化难度大
- 面向结构化网格的SpTRSV，可以利用网格信息进行并行和高效访存

$$\begin{bmatrix} L_{11} & & & & & & & \\ L_{21} & L_{22} & & & & & & \\ & L_{32} & L_{33} & & & & & \\ L_{41} & & & L_{44} & & & & \\ & L_{52} & & L_{54} & L_{55} & & & \\ & & L_{63} & & L_{65} & L_{66} & & \\ & & & L_{74} & & & L_{77} & \\ & & & & L_{85} & & L_{87} & L_{88} \\ & & & & & L_{96} & & L_{98} & L_{99} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \end{bmatrix}$$

2 传统的并行方法及其问题

- 通常该类问题采用Level-set方法进行并行，其每个level均需要进行同步，同步开销大，且会导致离散访存，性能偏低
- 还可利用Jacobi迭代的思想进行并行，并行度大幅提高但实际应用中通常需要数十次迭代，目前尚不实用



Algorithm 2: Fine-Grained Parallel Incomplete Factorization.

```
Set unknowns l_ij and u_ij to initial values
for sweep = 1,2,... until convergence do
    parallel for (i,j) ∈ S do
        if i > j then
            l_ij = (a_ij − Σ_{k=1}^{j−1} l_ik u_kj) / u_jj
        else
            u_ij = a_ij − Σ_{k=1}^{i−1} l_ik u_kj
        end
    end
end
```

3 算法及分析

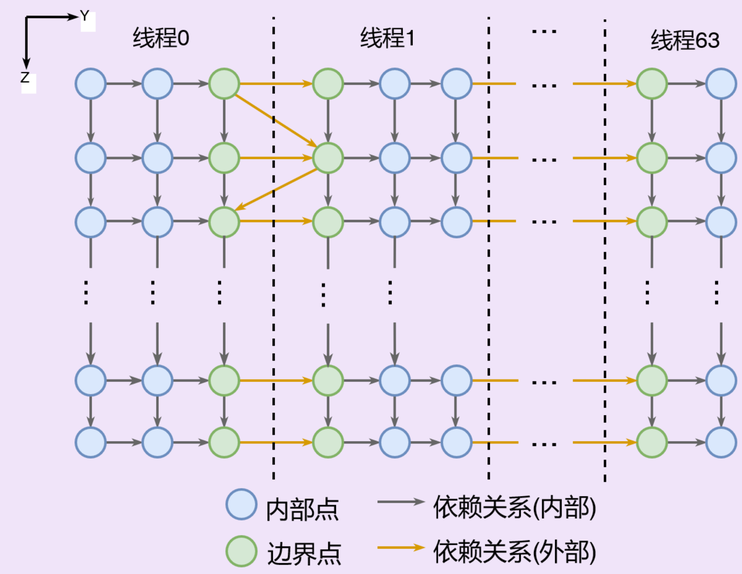
- 定义R(G)为结构网格中任意两个网格点的最大L1范数
- 算法中Ca只需访问一次，但可通过预处理将其排列成适用于申威平台的方式，提高访存带宽
- 重点优化B的访存

Algorithm 1: SpTRSV

```
for p = 0 to N − 1 do
    ω ← Ci[p]
    χ ← B[p]/Ca[ω]
    for q = 1 to Ci[p + 1] − Ci[p] − 1 do
        δ ← Cj[ω + q]
        B[δ] ← B[δ] − Ca[ω + q] · χ
    end
    X[p] ← χ
end
```

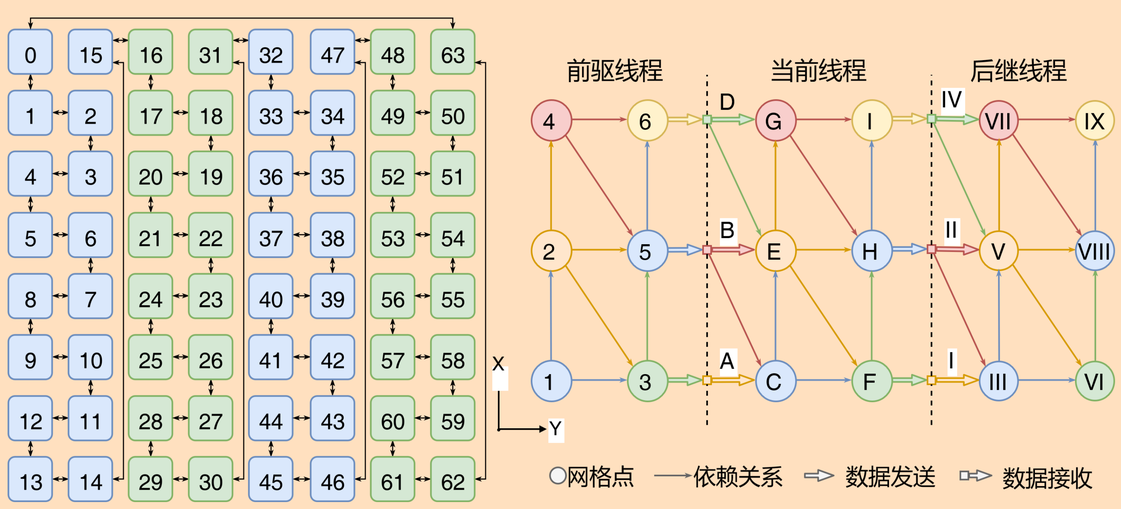
4 任务划分

- 面向国产申威处理器进行并行
- 沿y方向将网格切成64个柱状区域，每个从核负责一个
- 最小分块为R(G)，限制通信范围



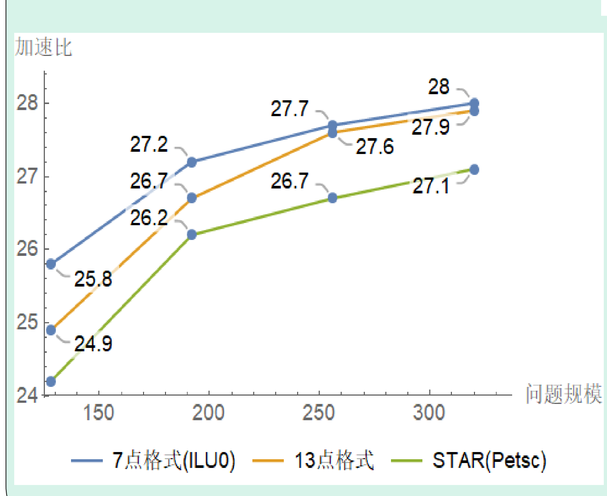
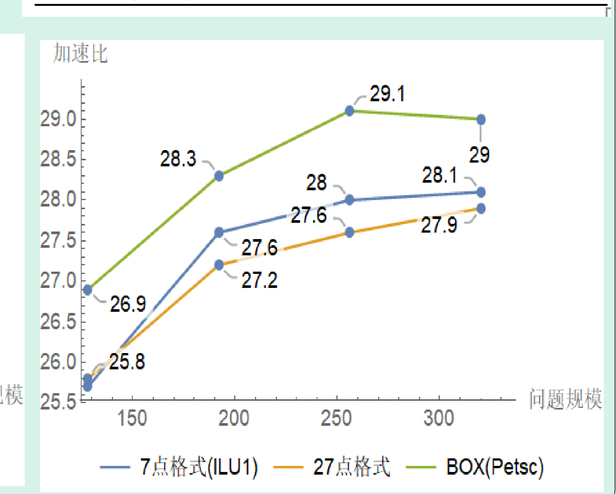
5 存储及通信

- B[p]使用循环队列存储，计算时通过寄存器通信交换
- 算法使用双向链式通信网络消除竞争
- 对计算和通信进行精心排布



6 实验结果

- 选用了常见的几种结构网格进行测试
- 带宽利用率最高可达94.5%
- 加速比24.6x-29.1x

表 2 性能测试的访存带宽(GB/s)数据

网格类型	128^3	192^3	256^3	320^3
7 点格式(ILU0)	20.41	21.80	22.60	22.92
13 点格式	21.17	22.37	22.81	23.03
STAR(Petsc)	21.23	22.73	23.41	23.50
7 点格式(ILU1)	21.87	23.41	23.87	23.93
27 点格式	22.55	23.78	24.24	24.43
BOX(Petsc)	22.82	23.94	24.51	24.57