当我们仰望星空
——并行计算星系模拟

敬请欣赏，宇宙中天体的舞蹈

从前，搬个小板凳
在那片夜空下仰望天空
如今，借助并行计算的力量
我们重现宇宙的奇妙与壮丽

PB21071489
黄俊善

目录

[01] 最终性能

GPU占用

95%+

20000规模加速比

42914倍

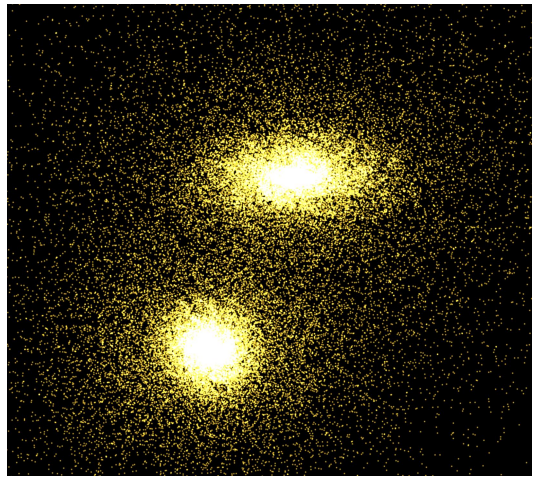
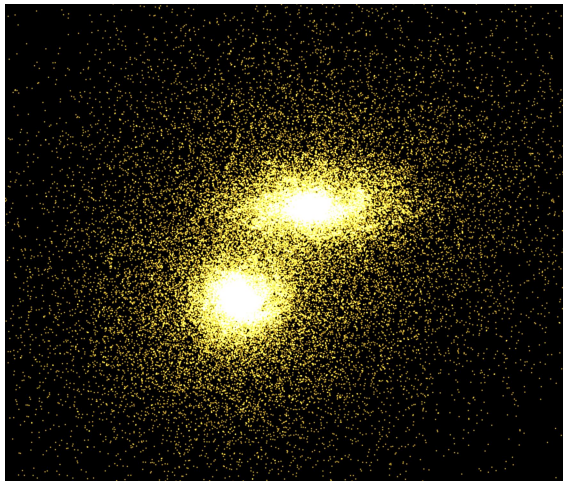
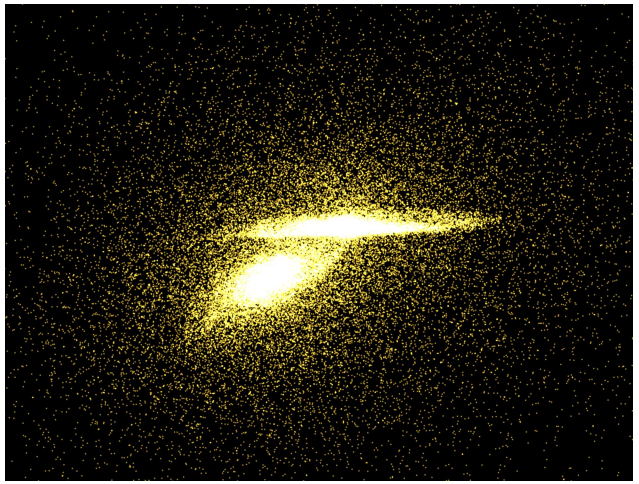
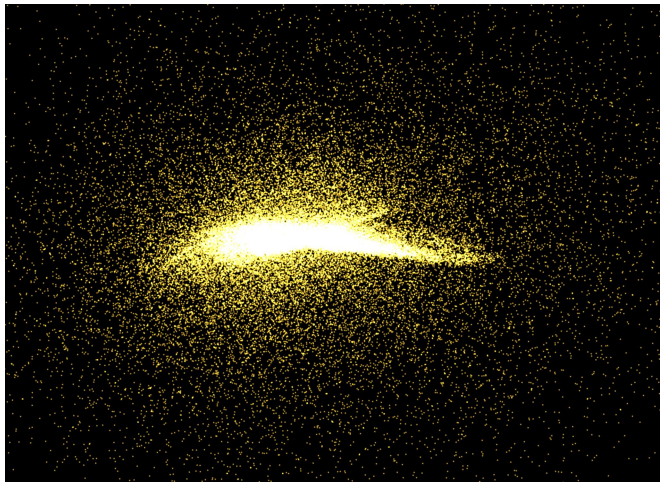
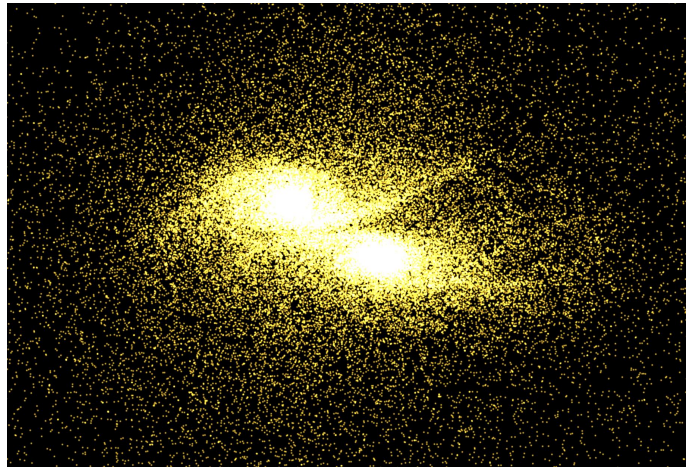
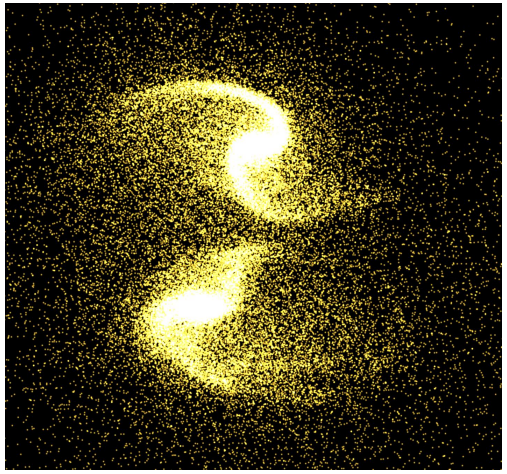
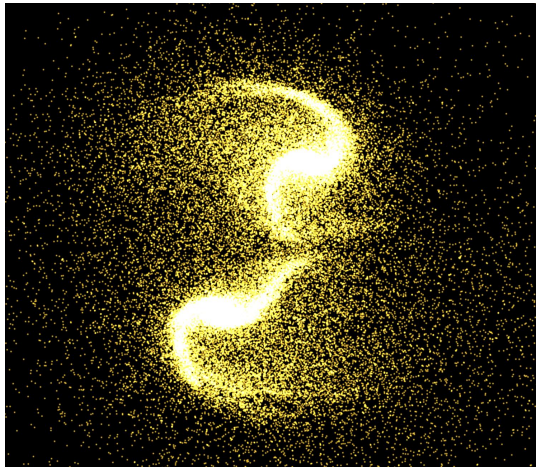
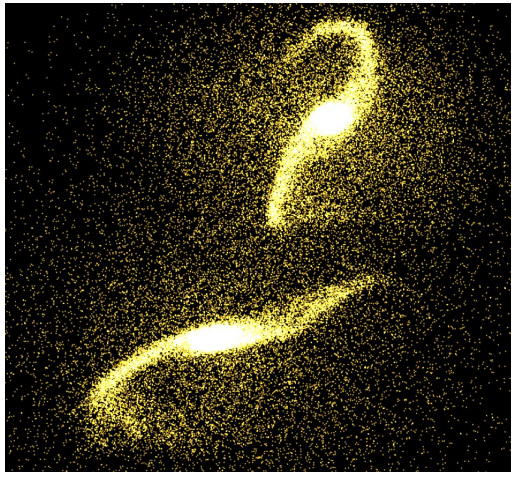
GPU功耗占用

99%+

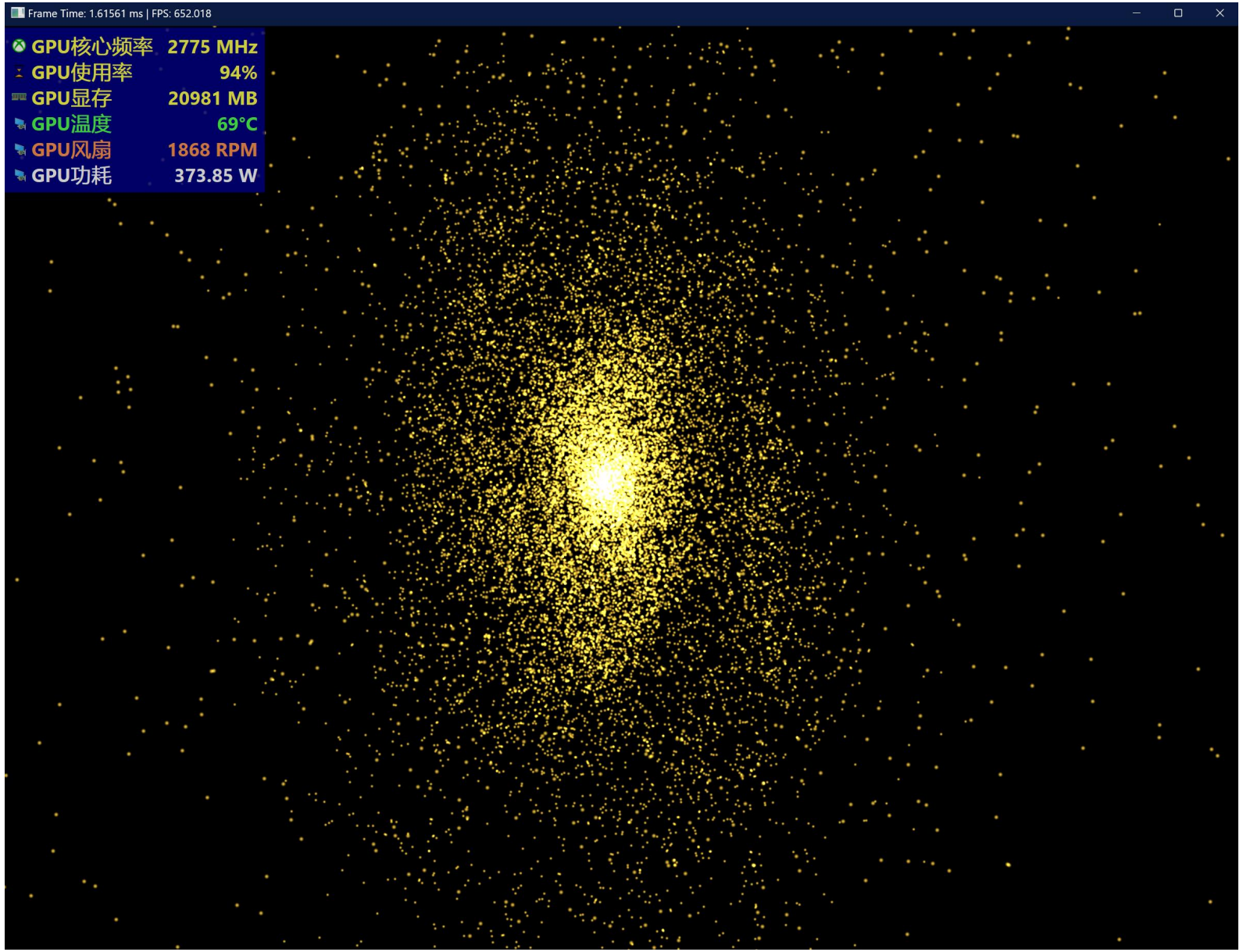
80000规模加速比

55039倍

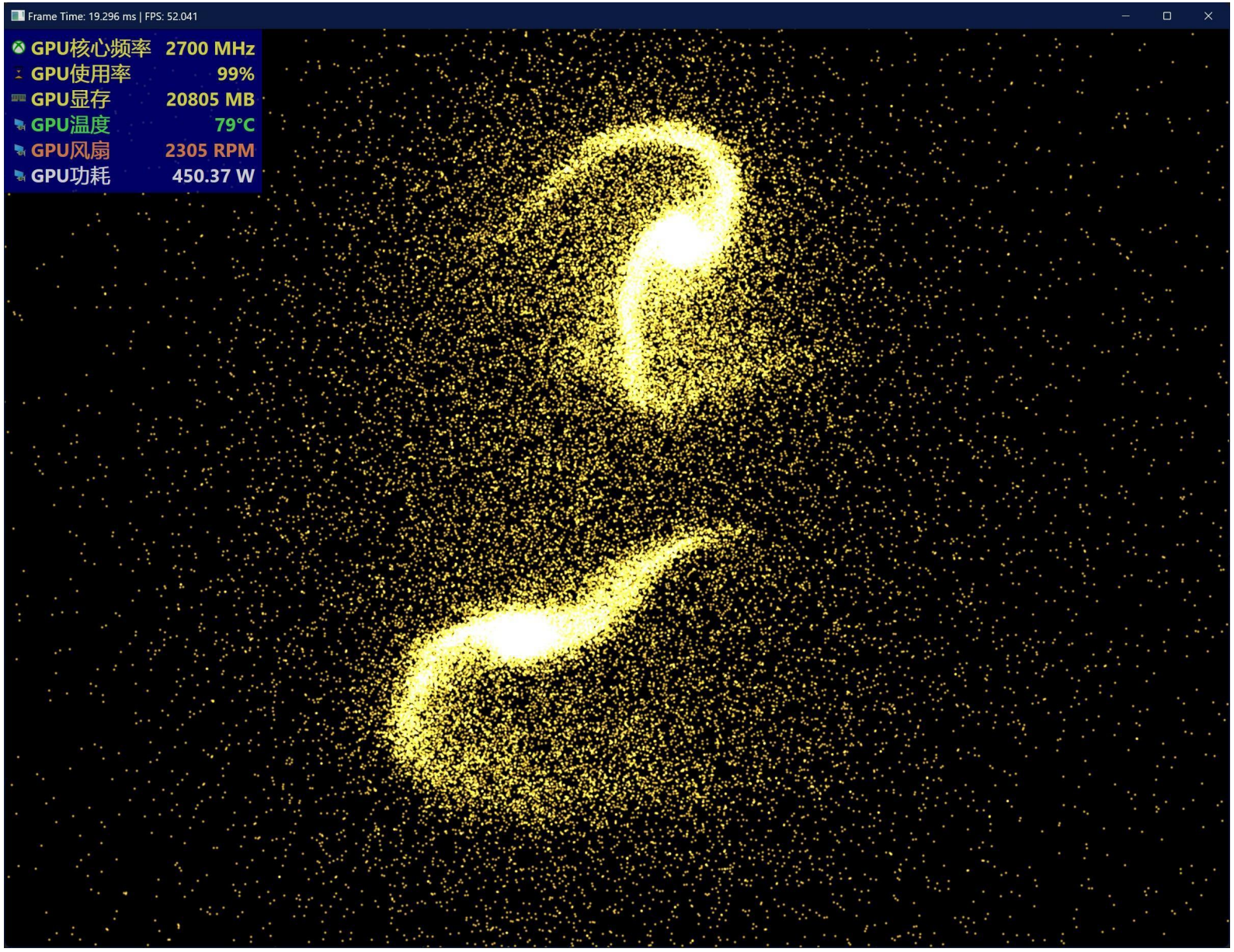
[01] 效果展示

[01] 效果展示


Frame Time: 1.61561 ms | FPS: 652.018
GPU核心频率 2775 MHz
GPU使用率 94%
GPU显存 20981 MB
GPU温度 69°C
GPU风扇 1868 RPM
GPU功耗 373.85 W

[01] 效果展示


Frame Time: 19.296 ms | FPS: 52.041
GPU核心频率 2700 MHz
GPU使用率 99%
GPU显存 20805 MB
GPU温度 79°C
GPU风扇 2305 RPM
GPU功耗 450.37 W

[Part 01]

背景与思路

[01]

宇宙，
如此壮丽

[01] 所有的所有，都来自于

$$\overrightarrow{F_G} = G \cdot \frac{M_1 M_2}{r^3} \cdot \overrightarrow{r}$$

万有引力公式

[01] 单位转换

太阳质量：1.988×10^30 kg
地球质量：5.972×10^24 kg
地日距离：1.496×10^11 m

$$\overrightarrow{F_G} = 6.67 \times 10^{-11} \cdot \frac{1.988 \times 10^{30} \times 5.972 \times 10^{24}}{(1.496 \times 10^{11})^3} \cdot \vec{r}$$

数值明显过大，不适合float等浮点表示 => 更换单位

[01] 单位转换

目标：

1. 平衡计算尺度以合理化利用浮点数的精度。
2. 通过单位的变化减少计算量。

- **时间单位：年**
  year/年，3.156x10^7s
- **距离单位：天文单位**
  AU/天文单位，1.496x10^11m
- **质量单位：太阳质量/39.478**
  M⊙/太阳质量，1.988x10^30kg

[01] 单位转换

时间单位：年
year/年，3.156x10^7s

距离单位：天文单位
AU/天文单位，1.496x10^11m

质量单位：太阳质量/39.478
M⊙/太阳质量，1.988x10^30kg

从而万有引力常数化为1：

$$G' = 6.6743 \times 10^{-11} \frac{m^3}{kg \cdot s^2} \times (\frac{1AU}{1.496 \times 10^{11} m})^3 \times (\frac{3.156 \times 10^7 s}{1年})^2 \times (\frac{1.98847 \times 10^{30} kg/39.478}{M\odot /39.478})$$

$$= 1(\frac{AU^3}{年^2 \cdot M\odot /39.478})$$

故，引力公式**减少一次**浮点运算！

$$\overrightarrow{F_G} = G \cdot \frac{M_1 M_2}{r^3} \cdot \overrightarrow{r} \xrightarrow{G=1} \overrightarrow{F_G} = \frac{M_1 M_2}{r^3} \cdot \overrightarrow{r}$$

[Part 02]

三种算法

[02] 模拟算法

计算各个天体之间相互作用

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

计算天体加速度

- 计算每个天体受到的合力
- 结合天体质量计算天体的加速度

更新天体位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

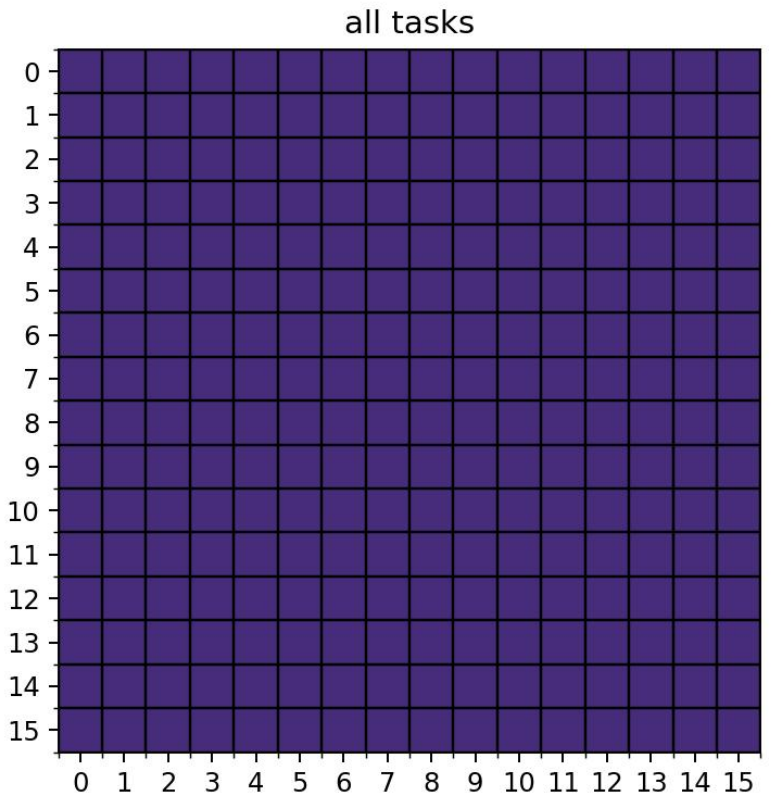
[02] 分析

计算各个天体之间相互作用

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

$$30 * O(n^2)$$

计算任务矩阵



[02] 分析

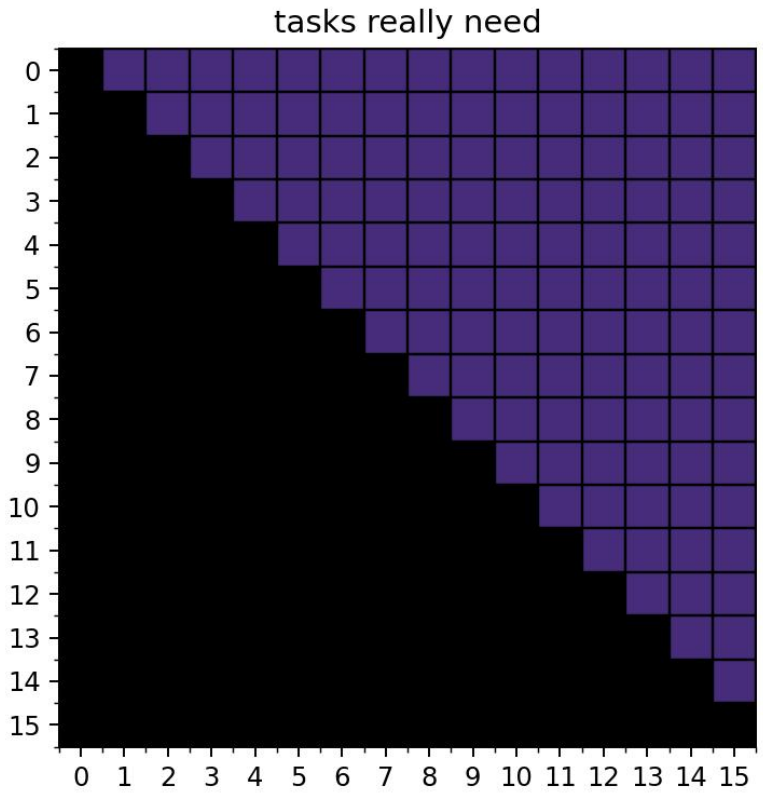
计算各个天体之间相互作用

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

$$30 * O(n^2)$$

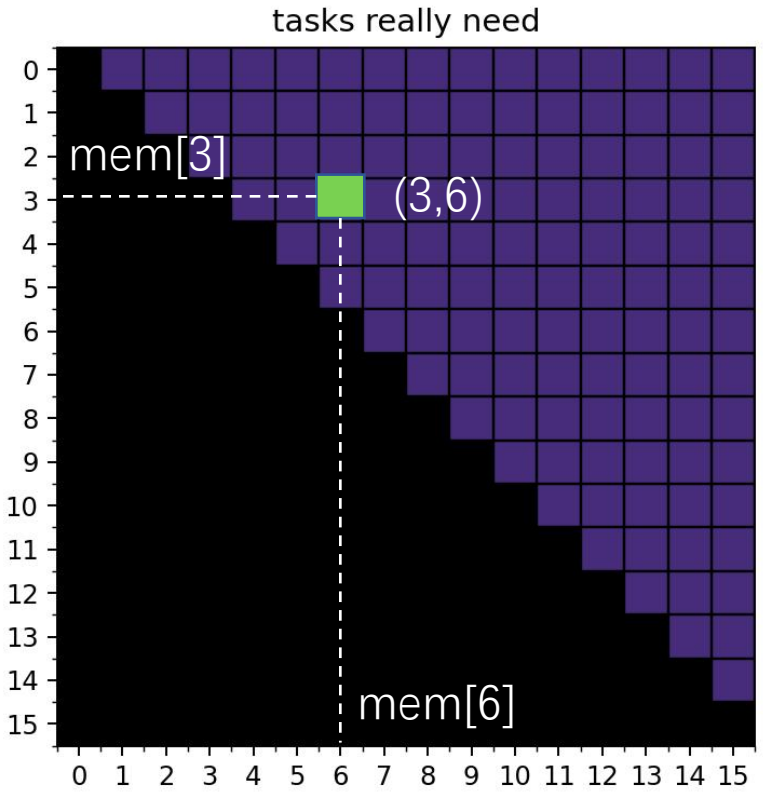
由于对称性，可以只计算一半

计算任务矩阵



[02] 最优分组方式探索

考虑到全局内存读写的延迟比较高，故在计算的时候应当先将全局数据保存到Block的共享内存中。

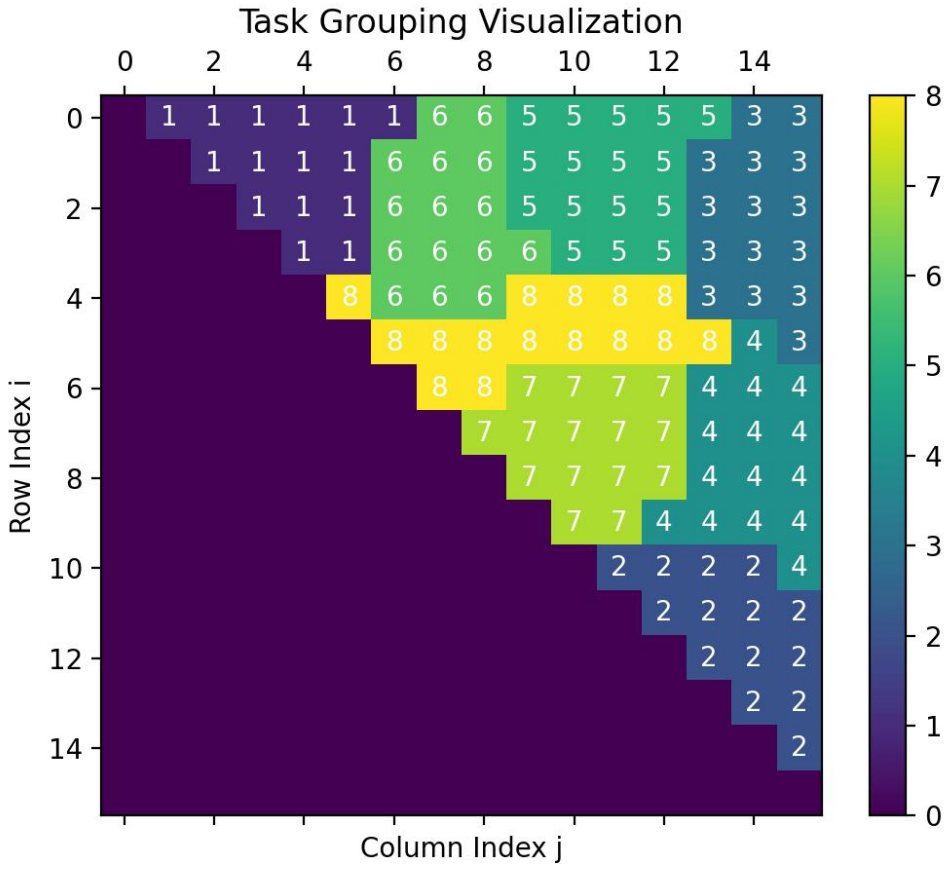


比如计算(3,6)这个任务，就需要加载3,6两个位置的内存。

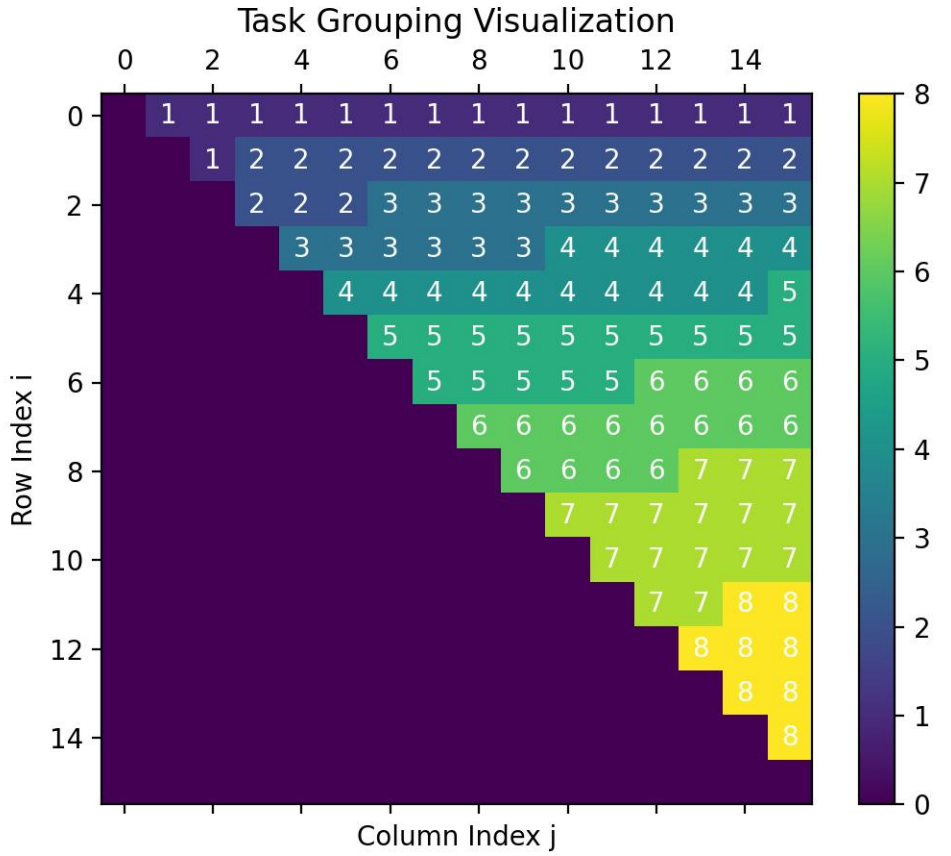
像这样一次加载了两个数据，我们就记内存加载量为2。

[02] 最优分组方式探索

考虑到全局内存读写的延迟比较高，故在计算的时候应当先将全局数据保存到Block的共享内存中。



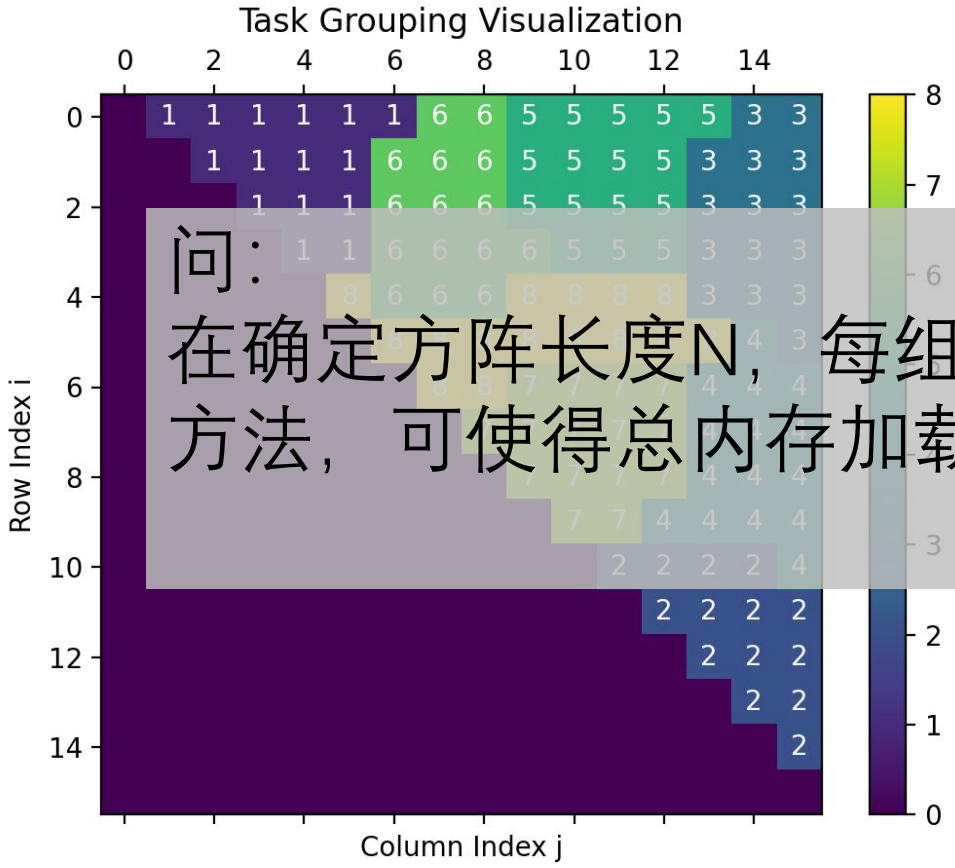
内存加载量=67



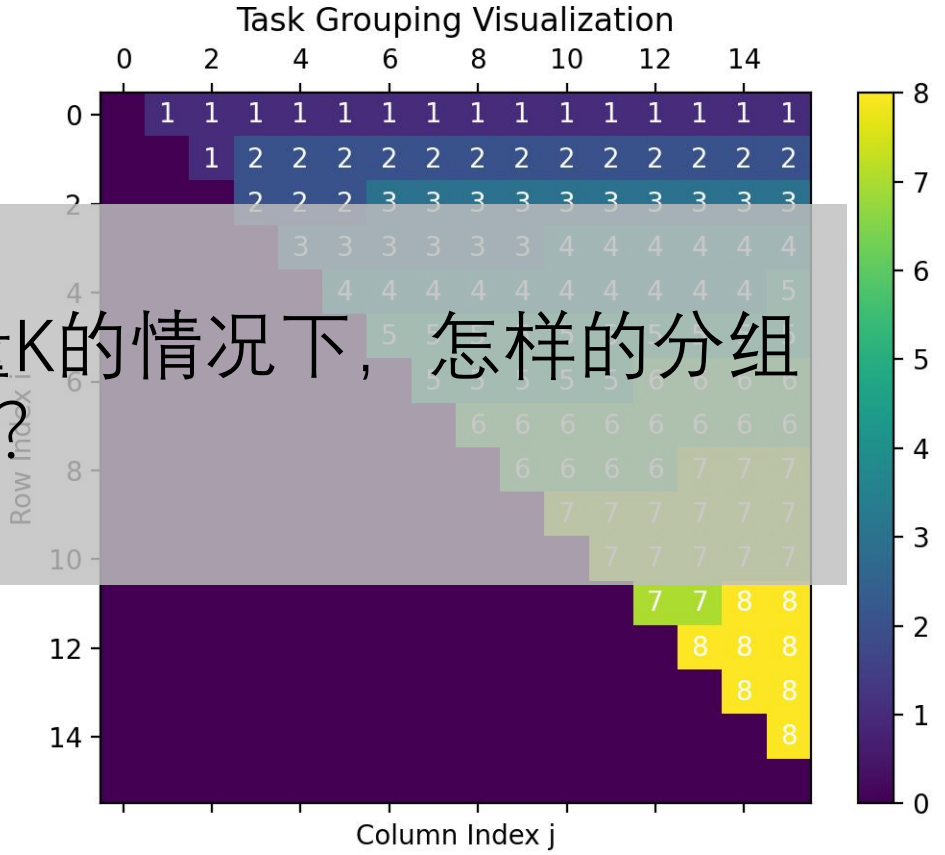
内存加载量=93

[02] 最优分组方式探索

考虑到全局内存读写的延迟比较高，故在计算的时候应当先将全局数据保存到Block的共享内存中。



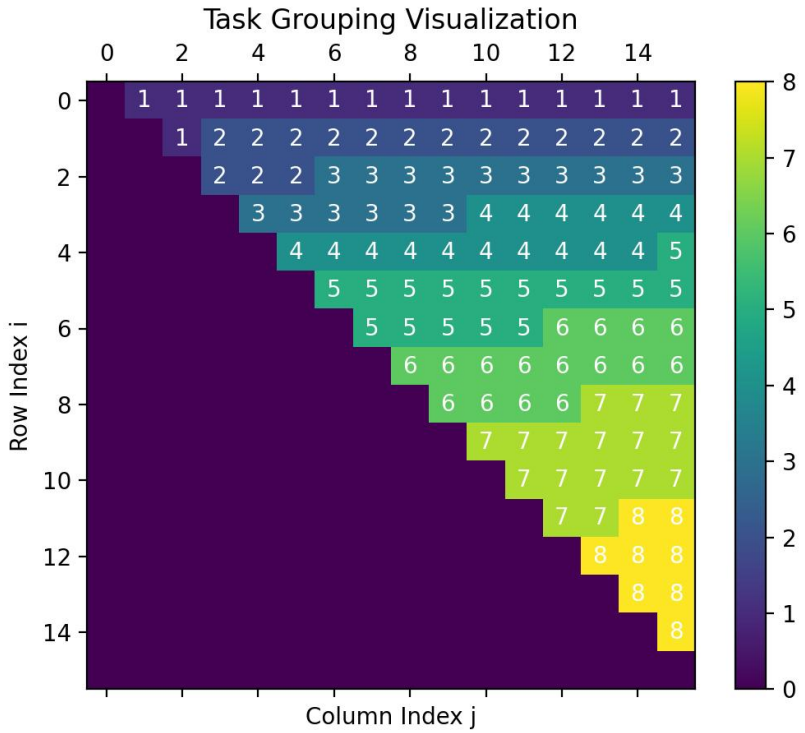
内存加载量=67



内存加载量=93

问：
在确定方阵长度N，每组元素数量K的情况下，怎样的分组方法，可使得总内存加载量L最小？

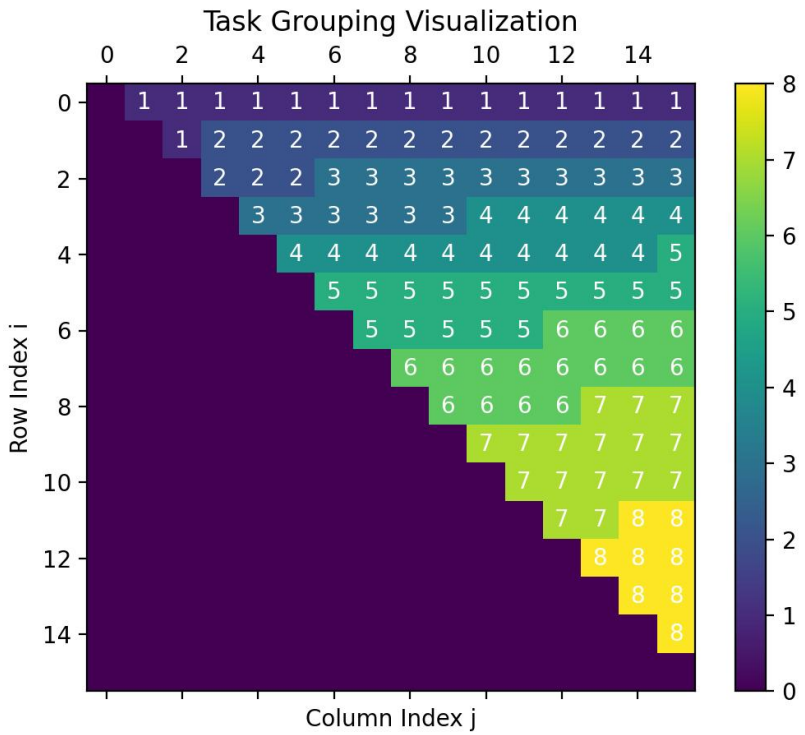
[02] 最优分组方式探索



贪心算法

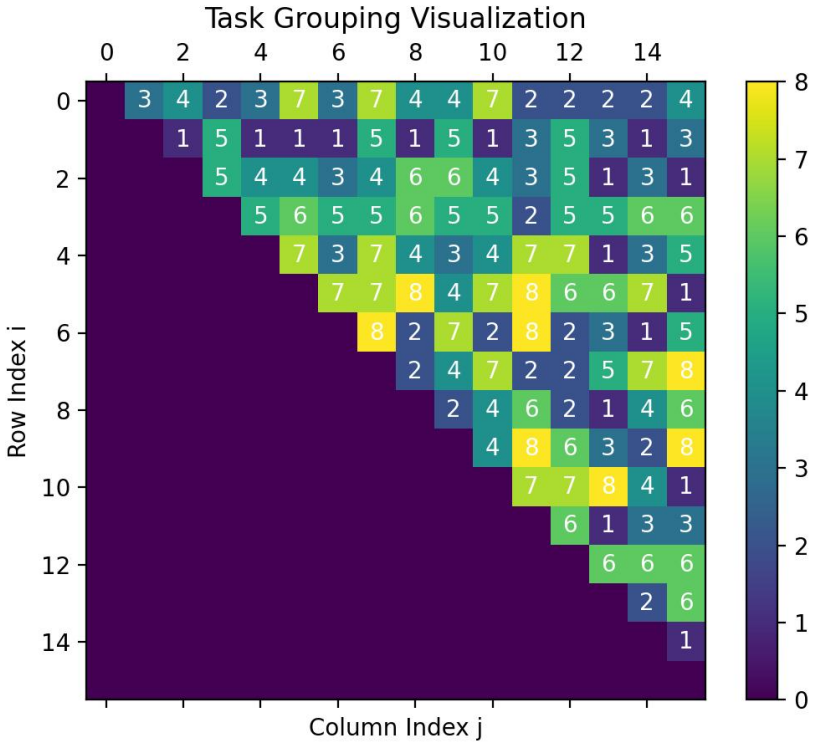
内存加载量：93

[02] 最优分组方式探索



贪心算法

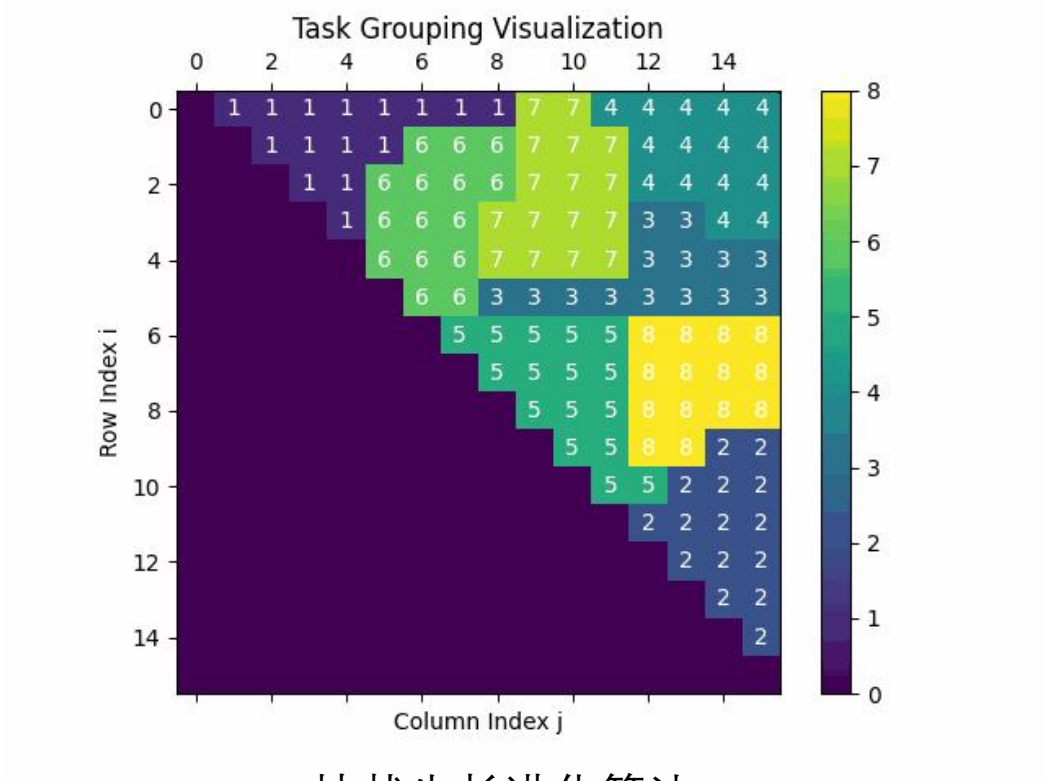
内存加载量：93



有规则的随机算法

内存加载量：82

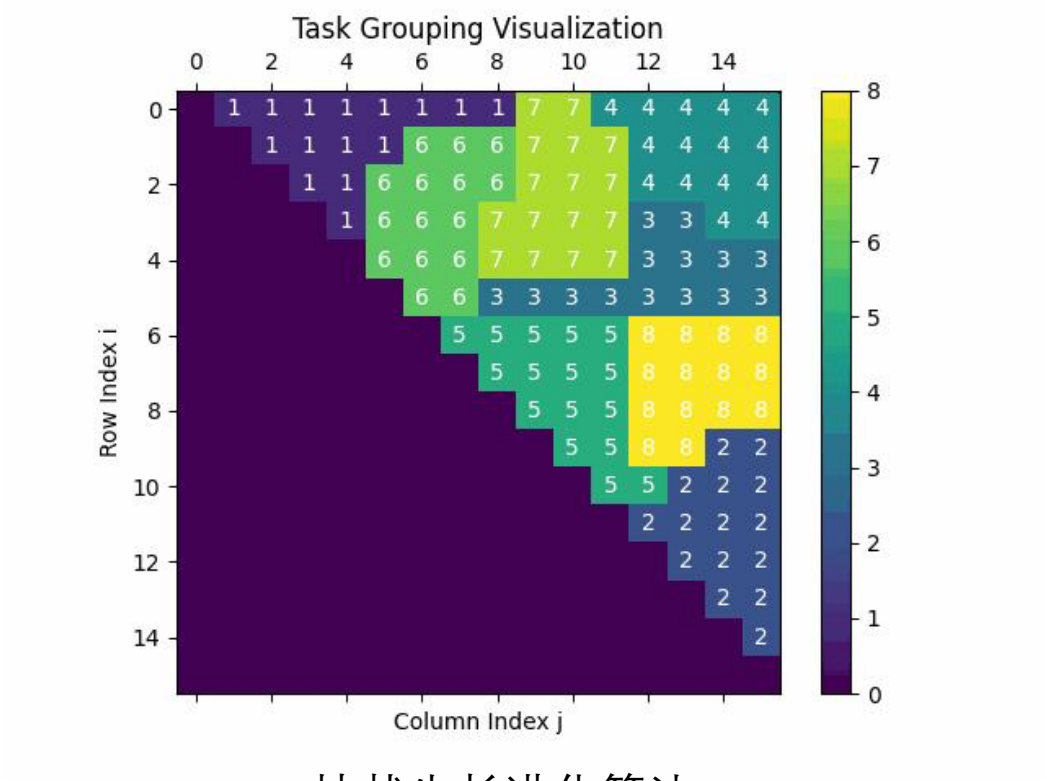
[02] 最优分组方式探索



块状生长进化算法

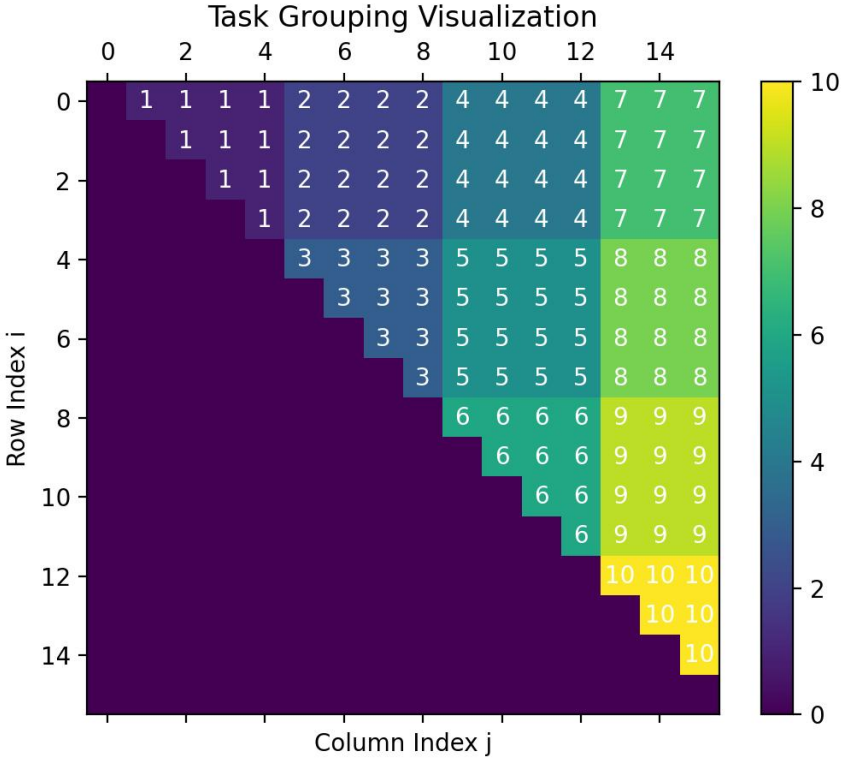
内存加载量：66

[02] 最优分组方式探索



块状生长进化算法

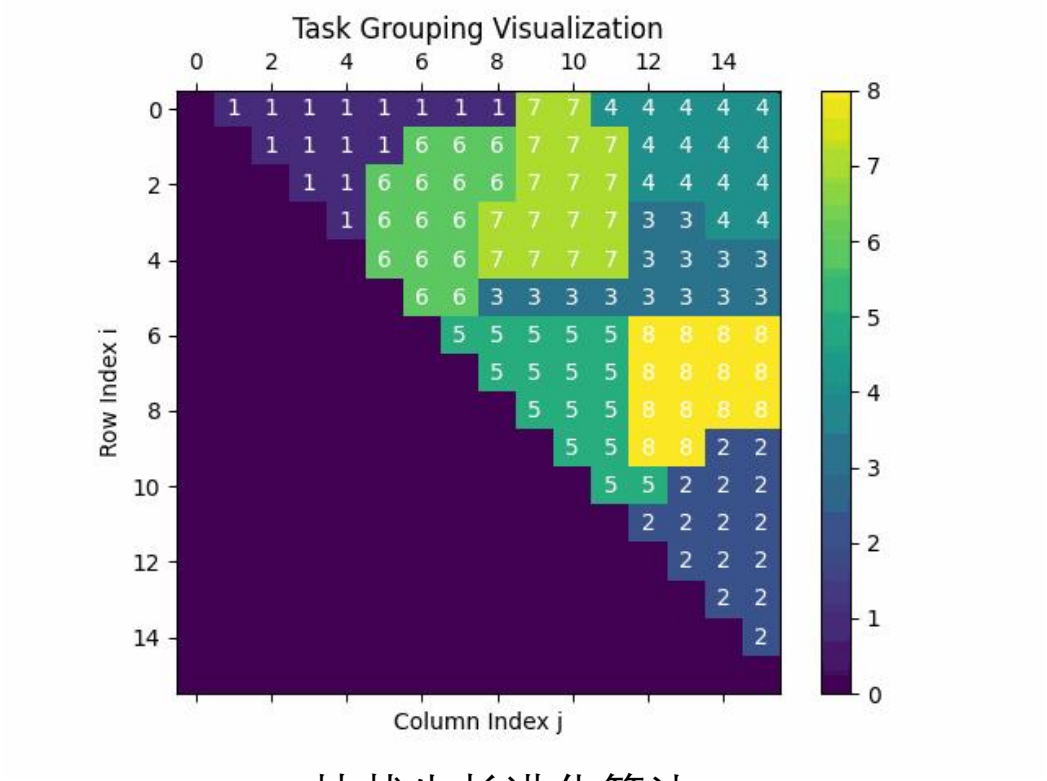
内存加载量：66



规则块状划分

内存加载量：64

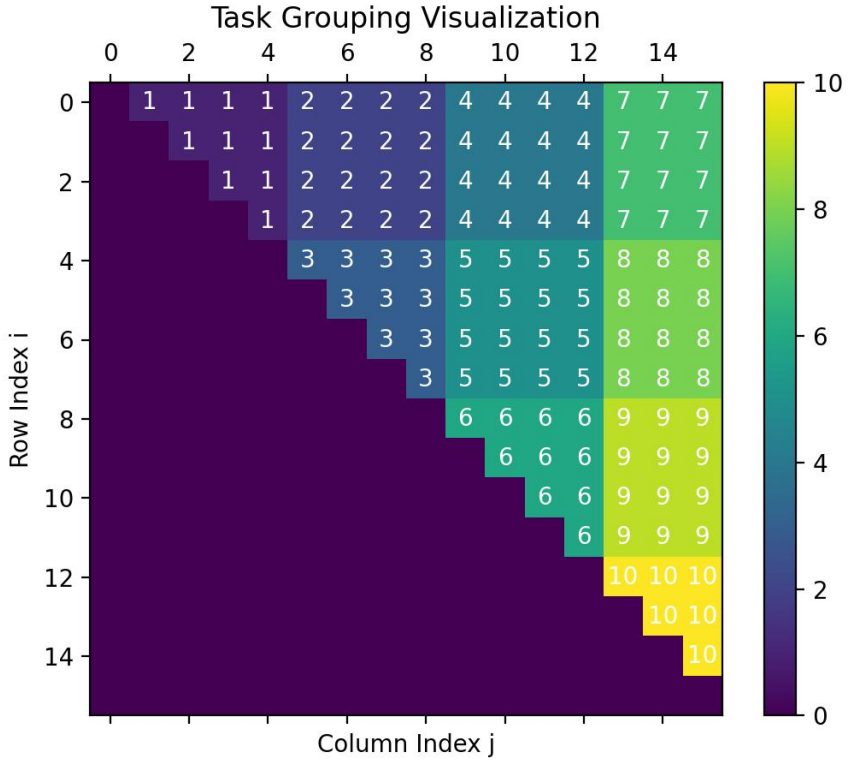
[02] 最优分组方式探索



块状生长进化算法

内存加载量：66

简洁且最优！



规则块状划分

内存加载量：64

[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

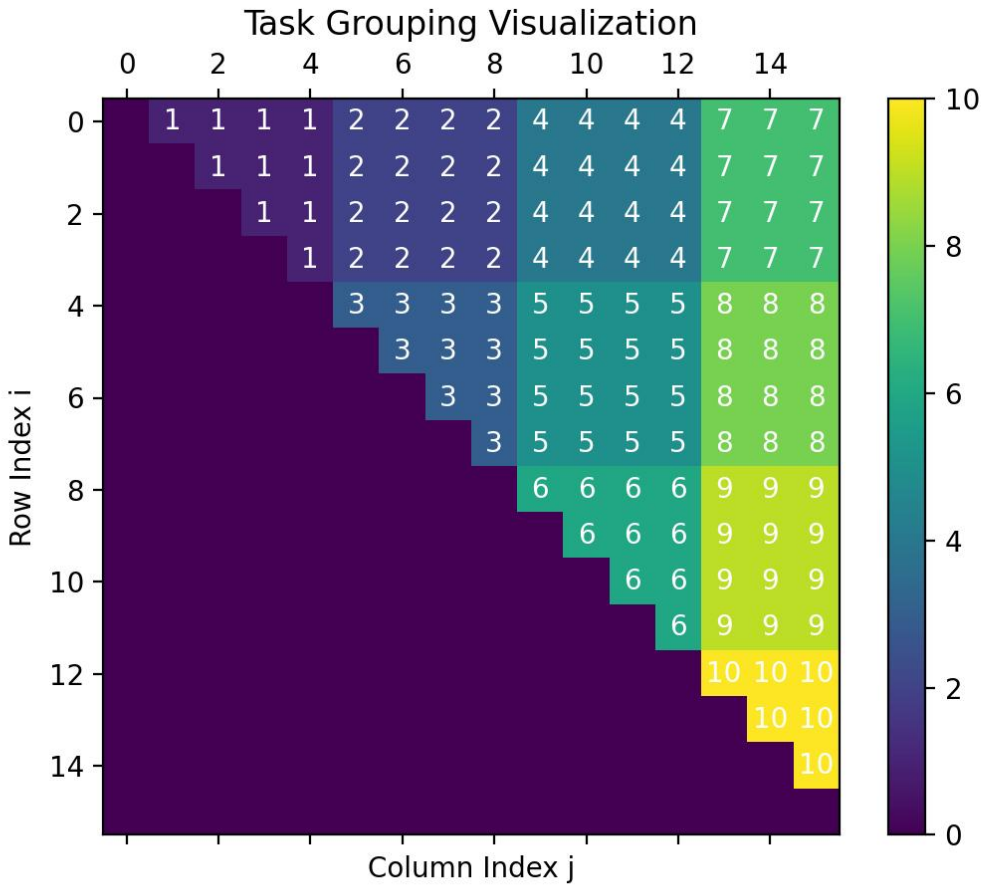
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算任务矩阵



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

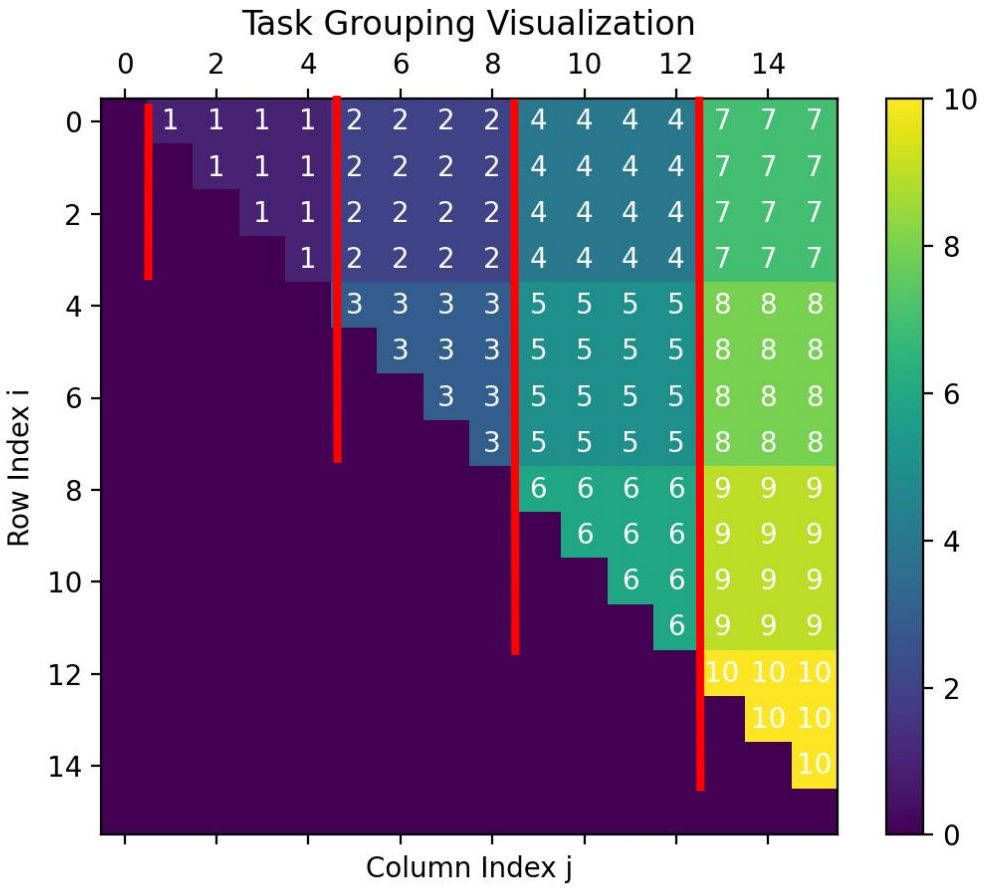
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

共享内存加载



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

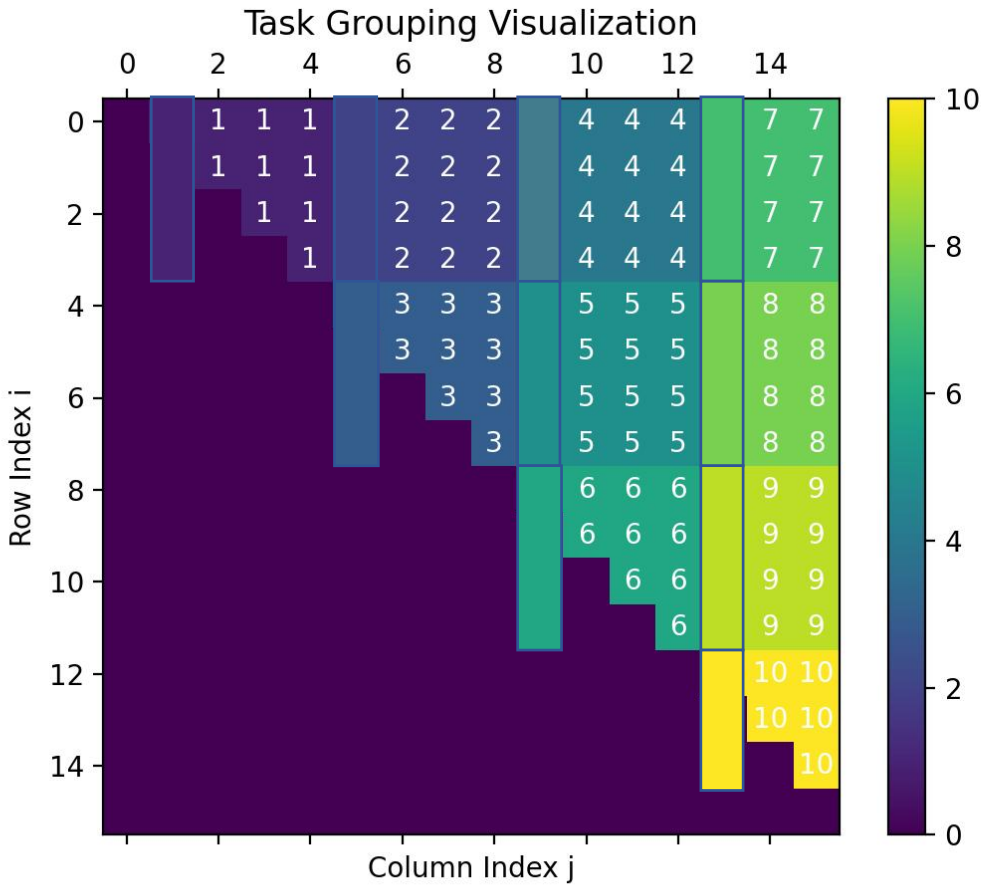
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算任务矩阵



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

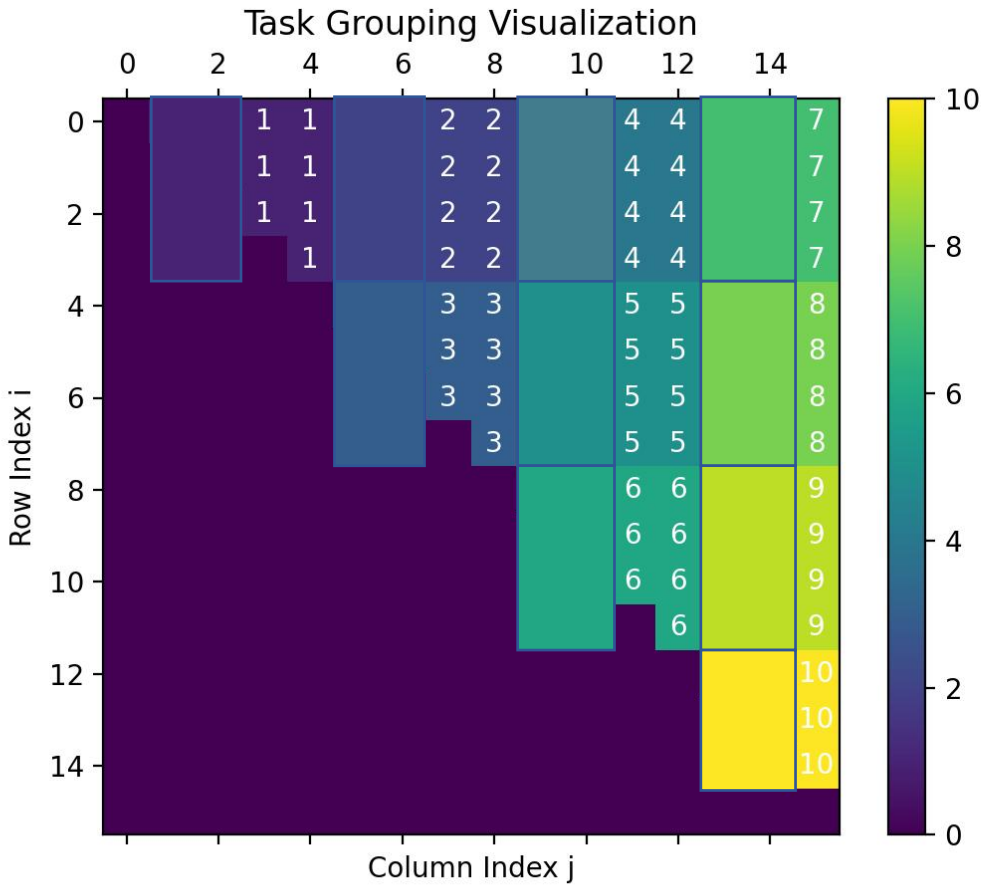
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算任务矩阵



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

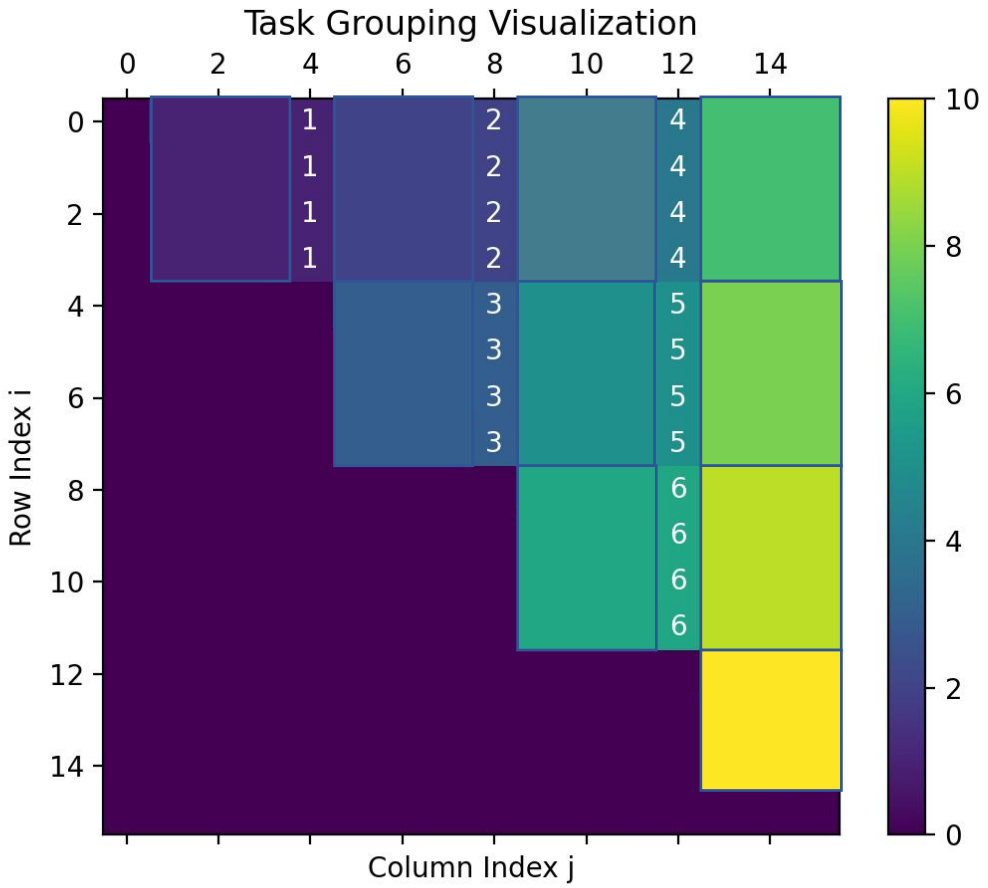
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算任务矩阵



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

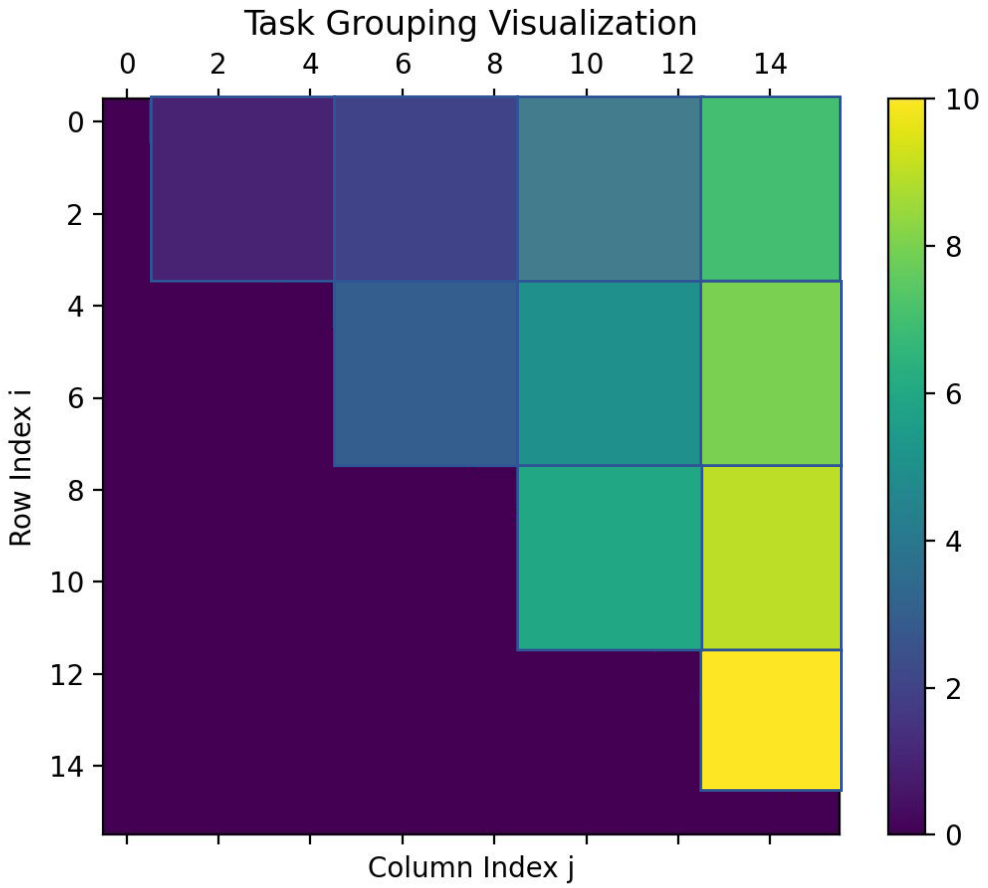
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算任务矩阵



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

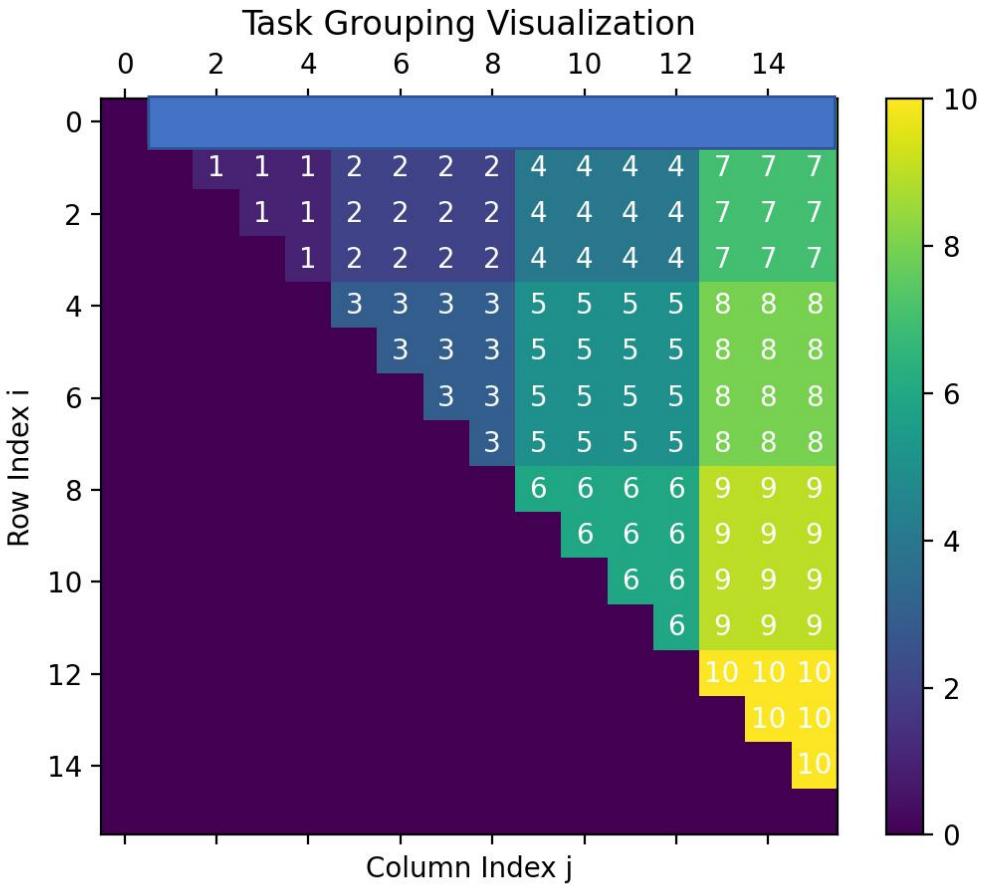
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体0收到的总引力G[0]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

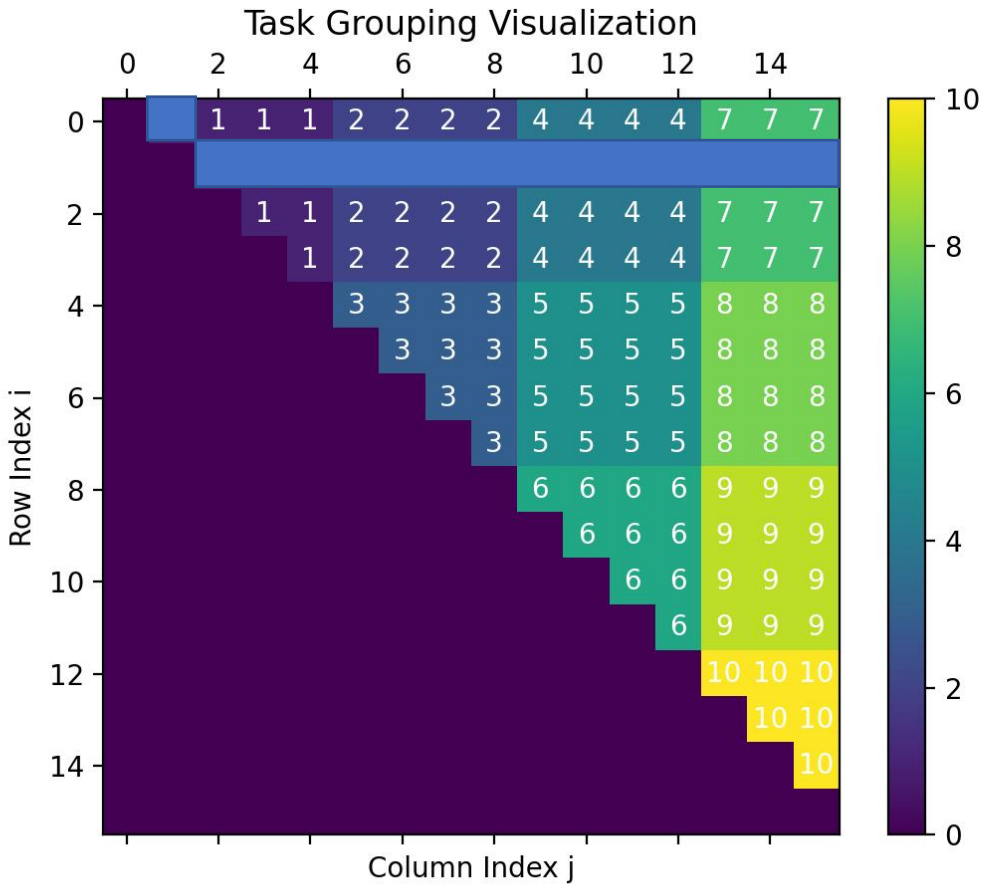
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体1收到的总引力G[1]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

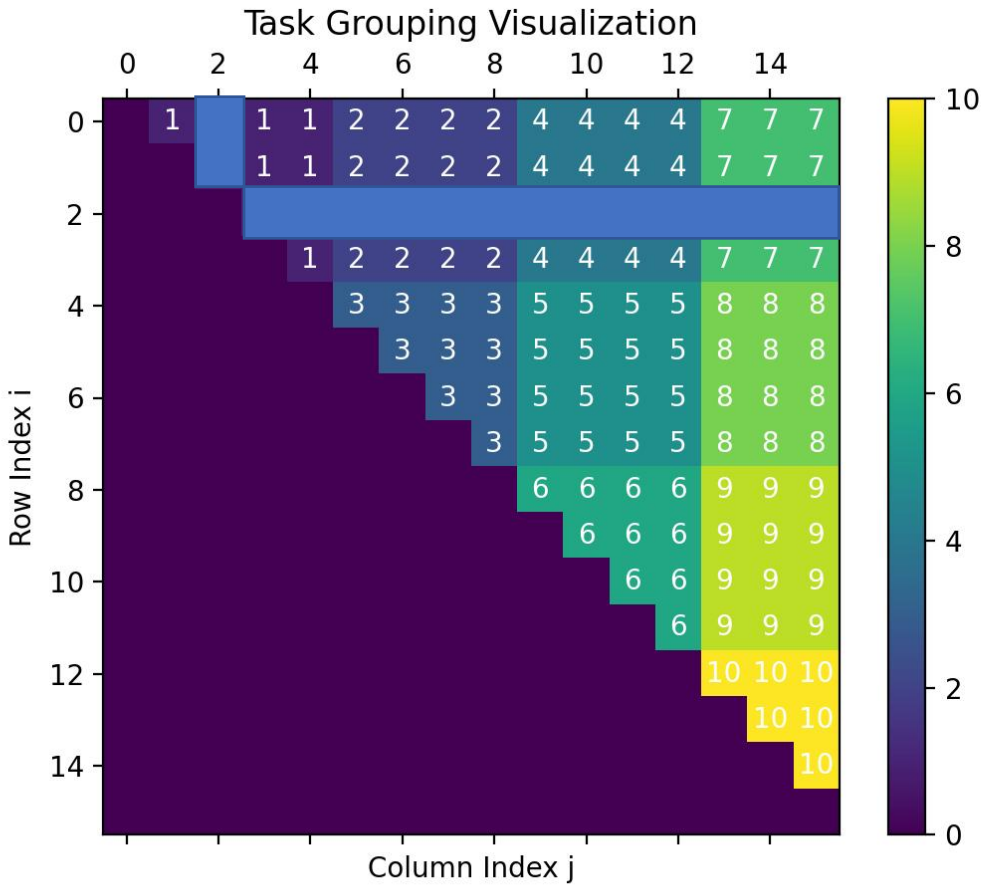
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体2收到的总引力G[2]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

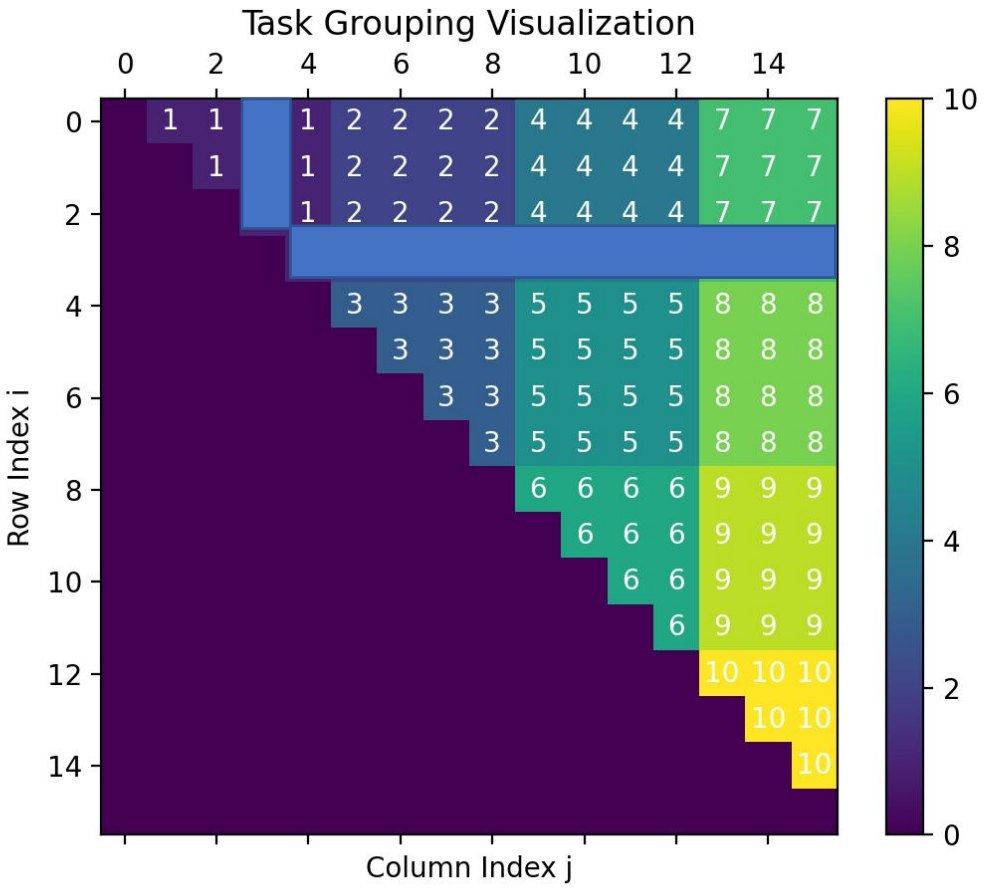
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体3收到的总引力G[3]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

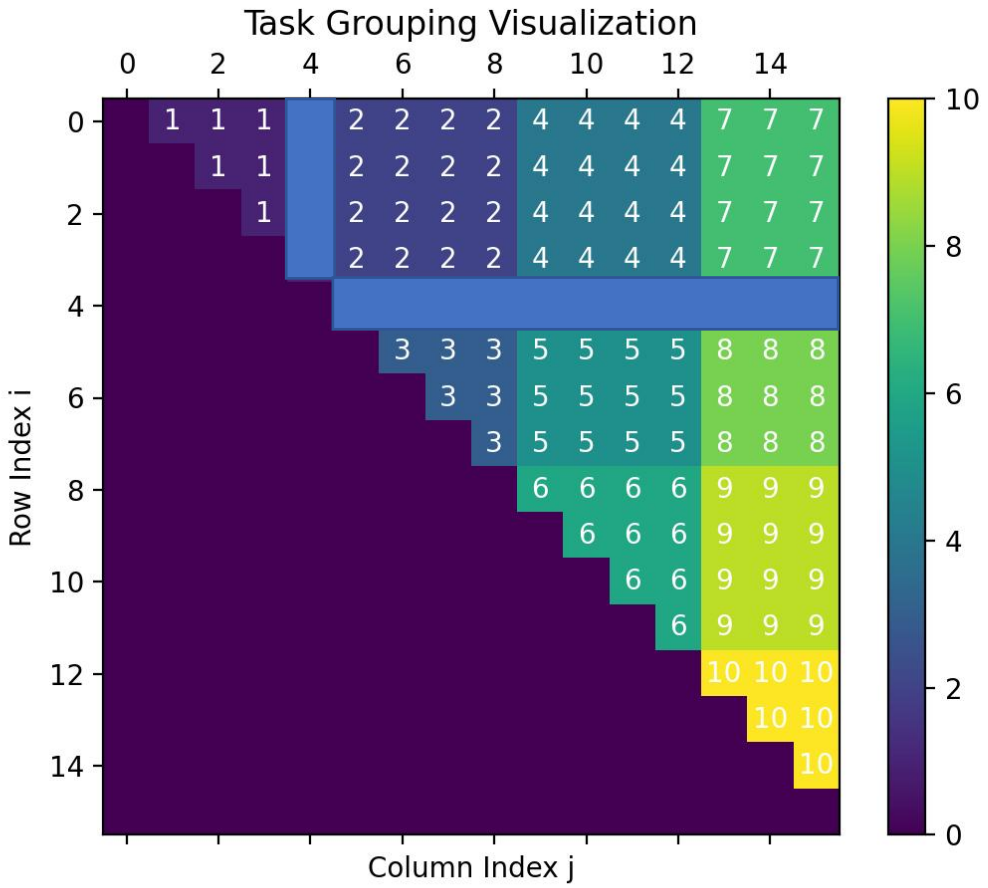
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体4收到的总引力G[4]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

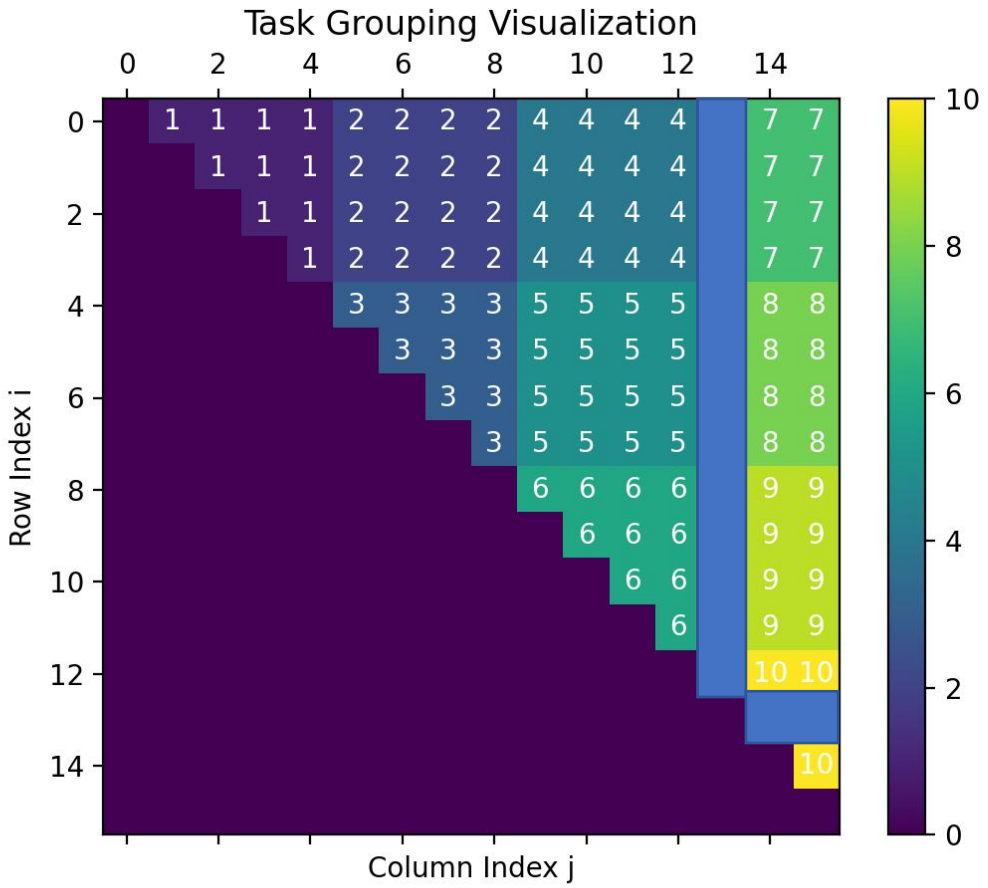
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体13收到的总引力G[13]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

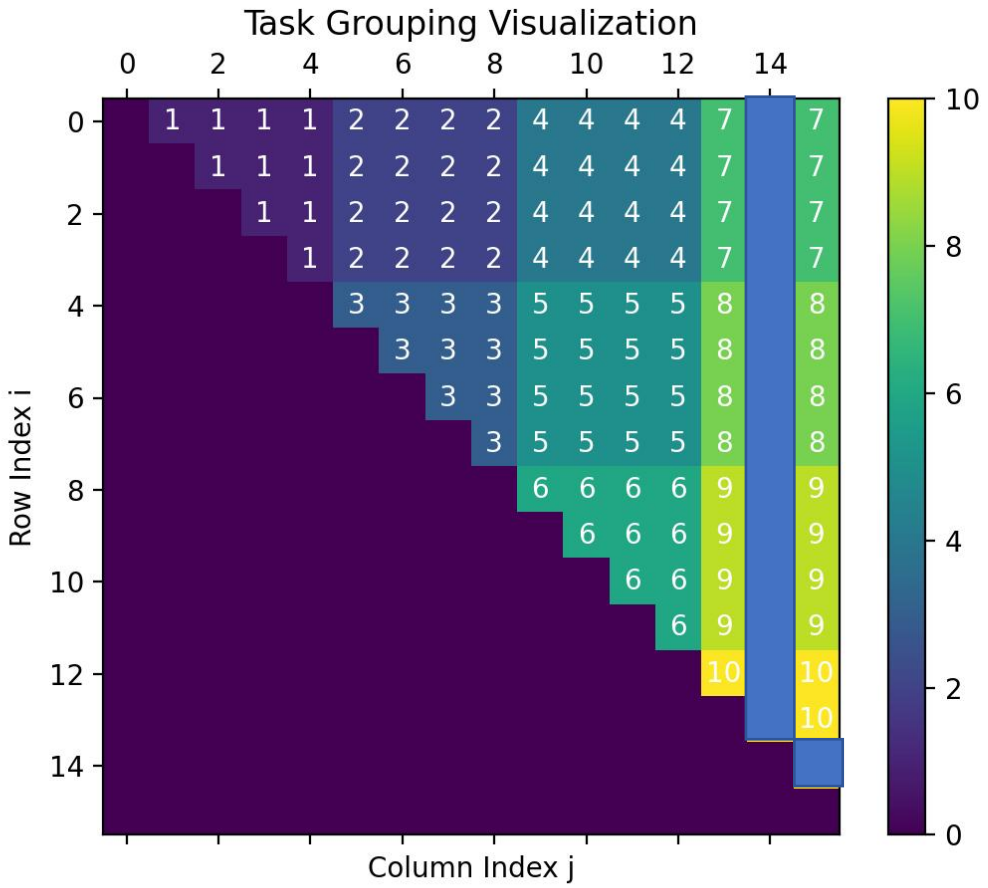
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体14收到的总引力G[14]



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

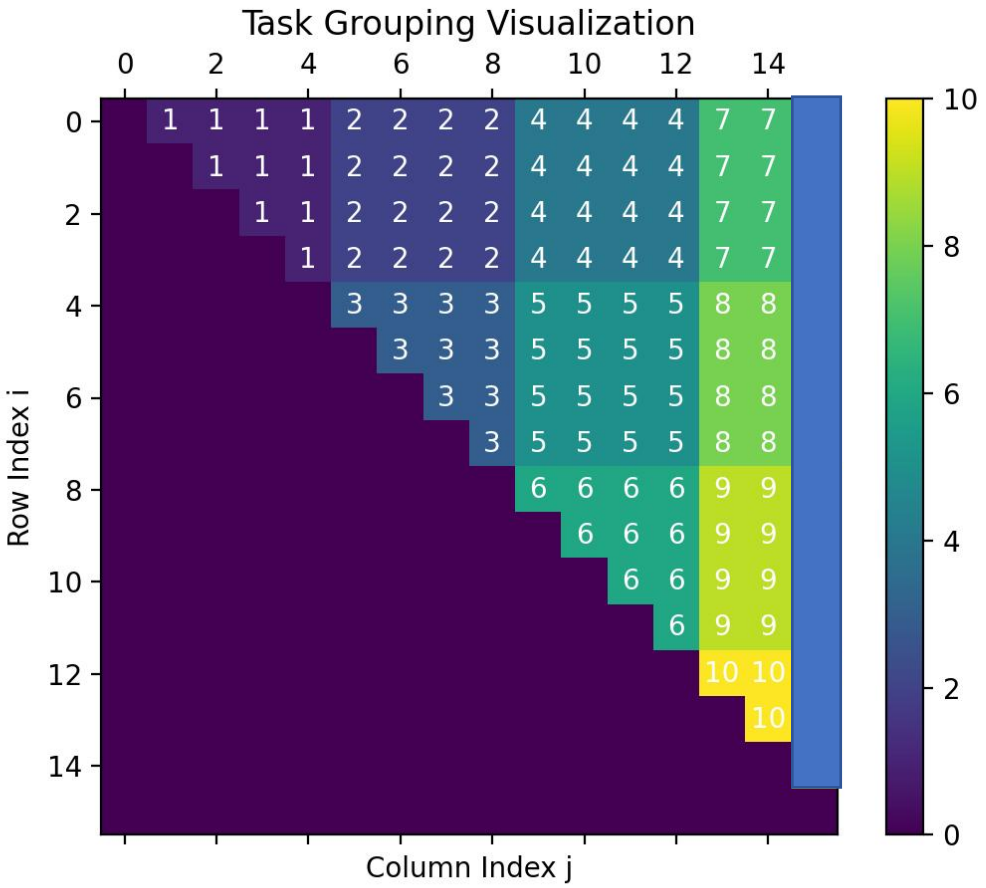
2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

计算天体15收到的总引力G[15]



[02] 方法A

1.计算各个天体之间引力

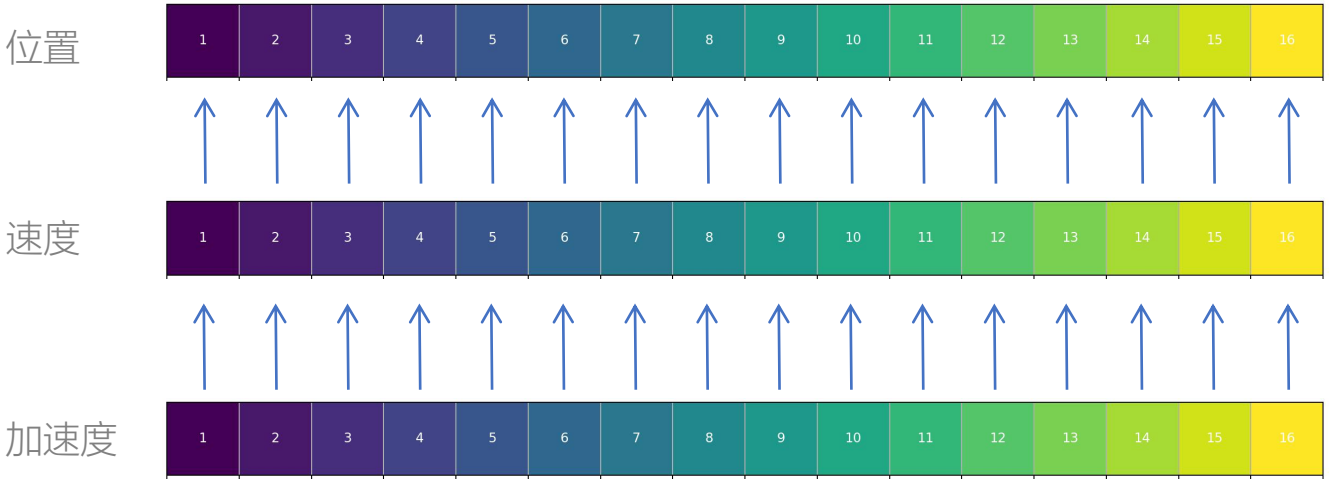
- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



[02] 方法A

1.计算各个天体之间引力

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算所有的引力

9ms

每次计算都要访问全局内存保存结果，太慢!

2. 计算合力

- 加和计算每个天体受到的合力
- 结合天体质量计算天体的加速度

10ms

（线程内串行加和）

从全局内存读取并线性加和，太慢!

3. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

0.3ms

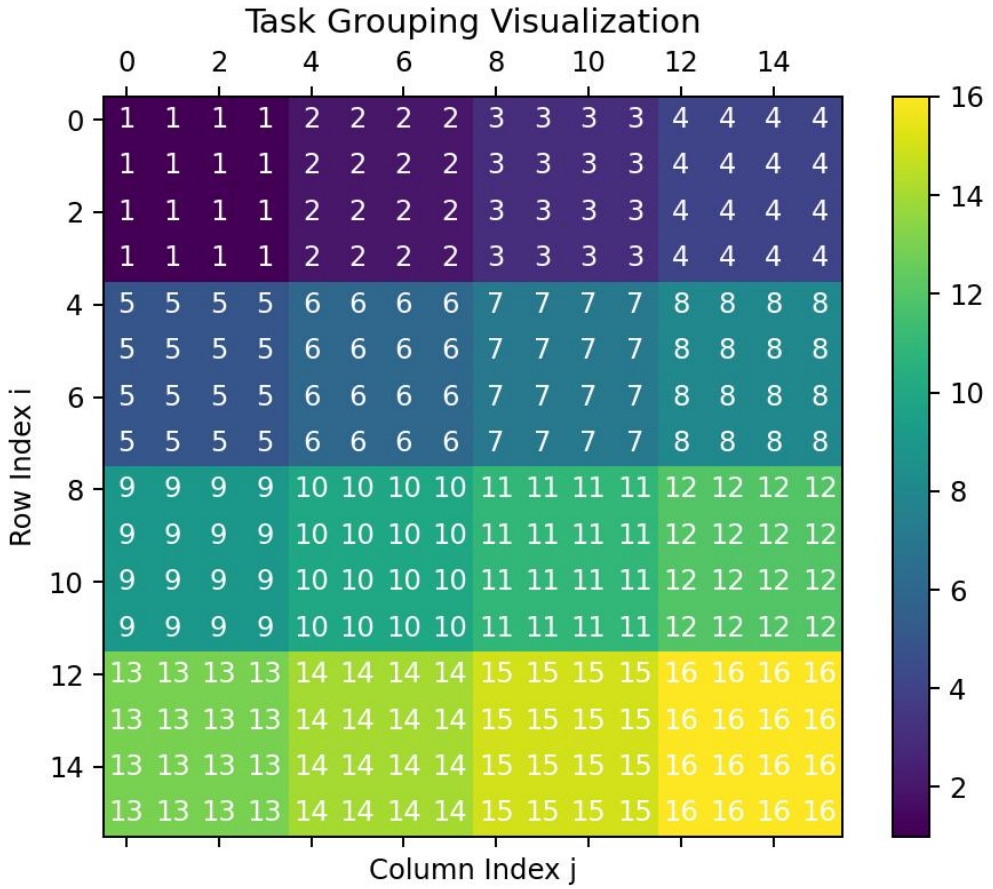
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



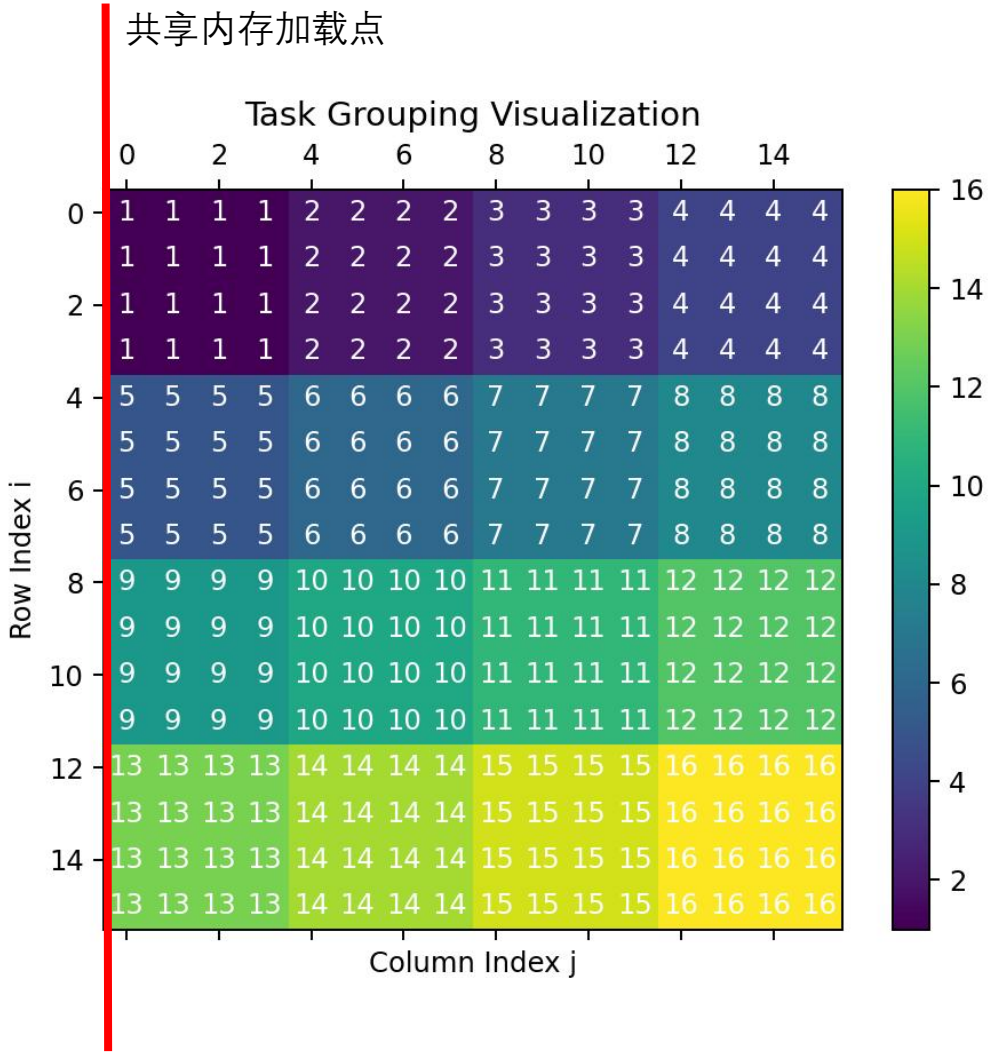
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



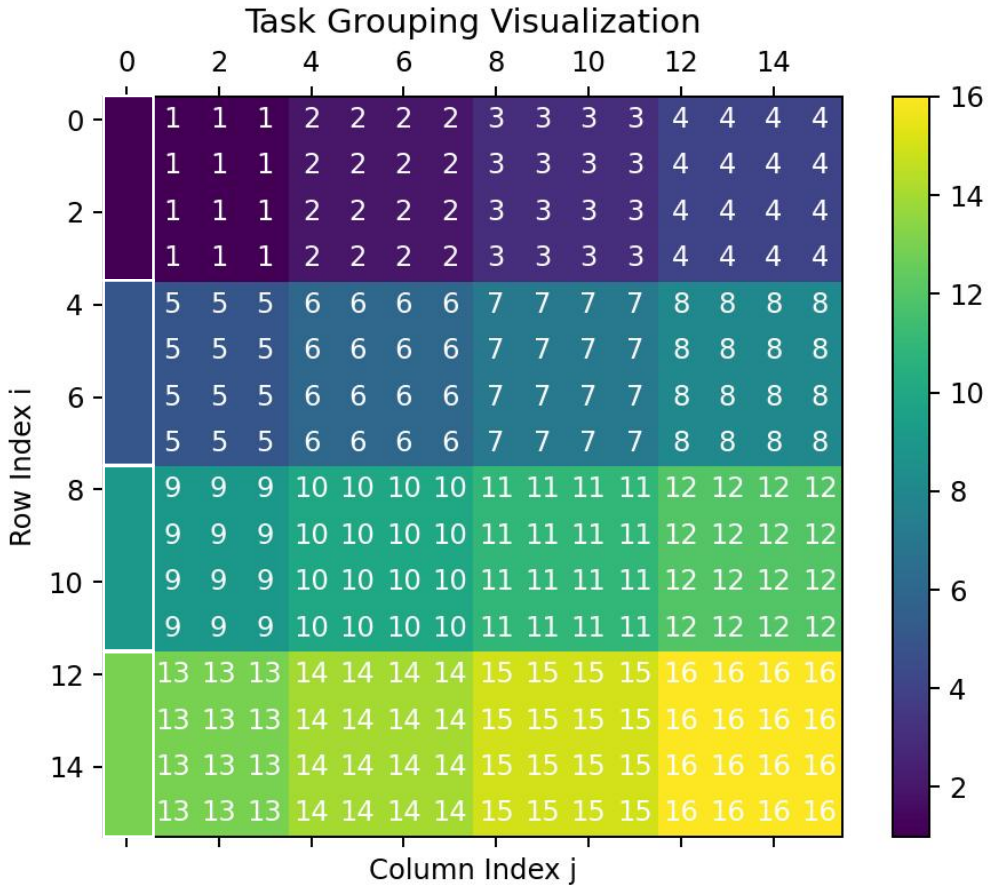
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



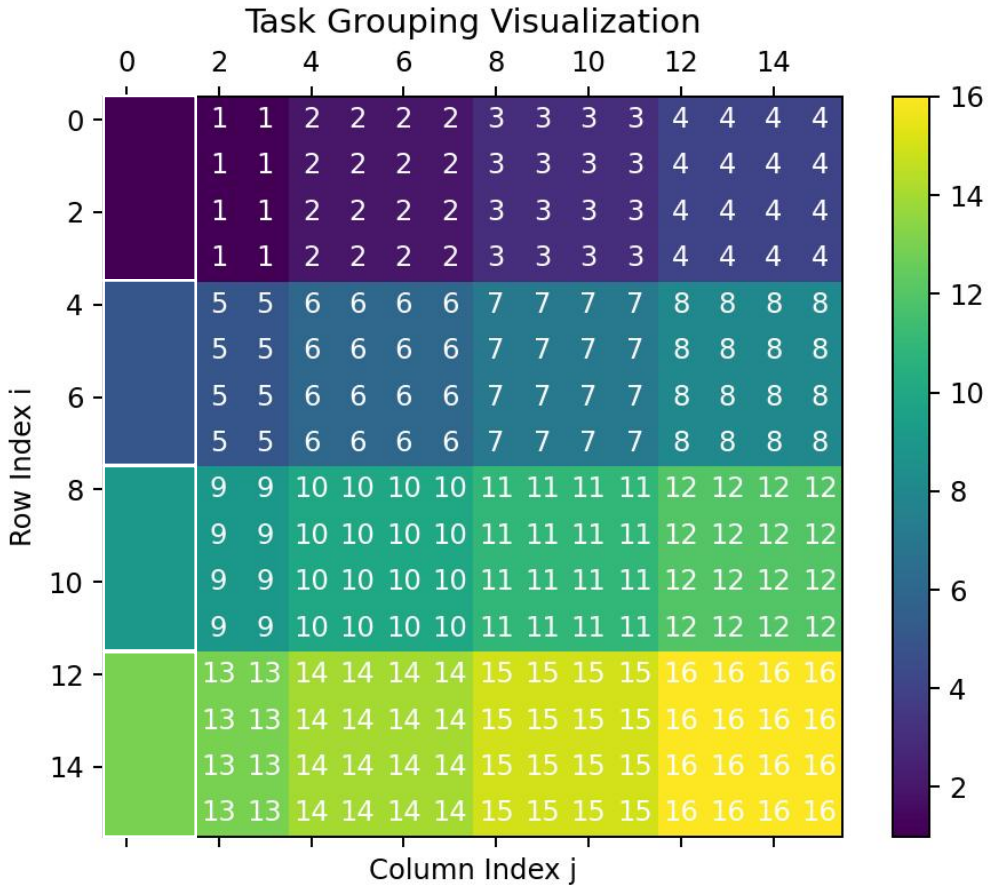
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



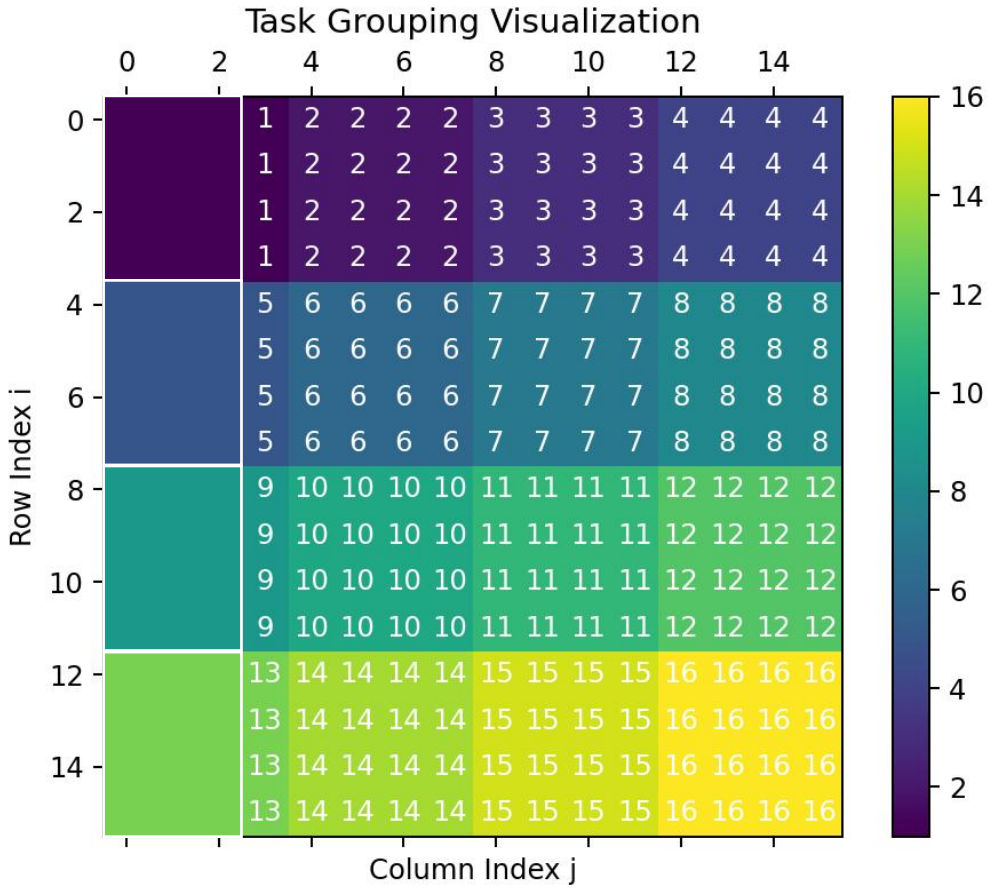
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



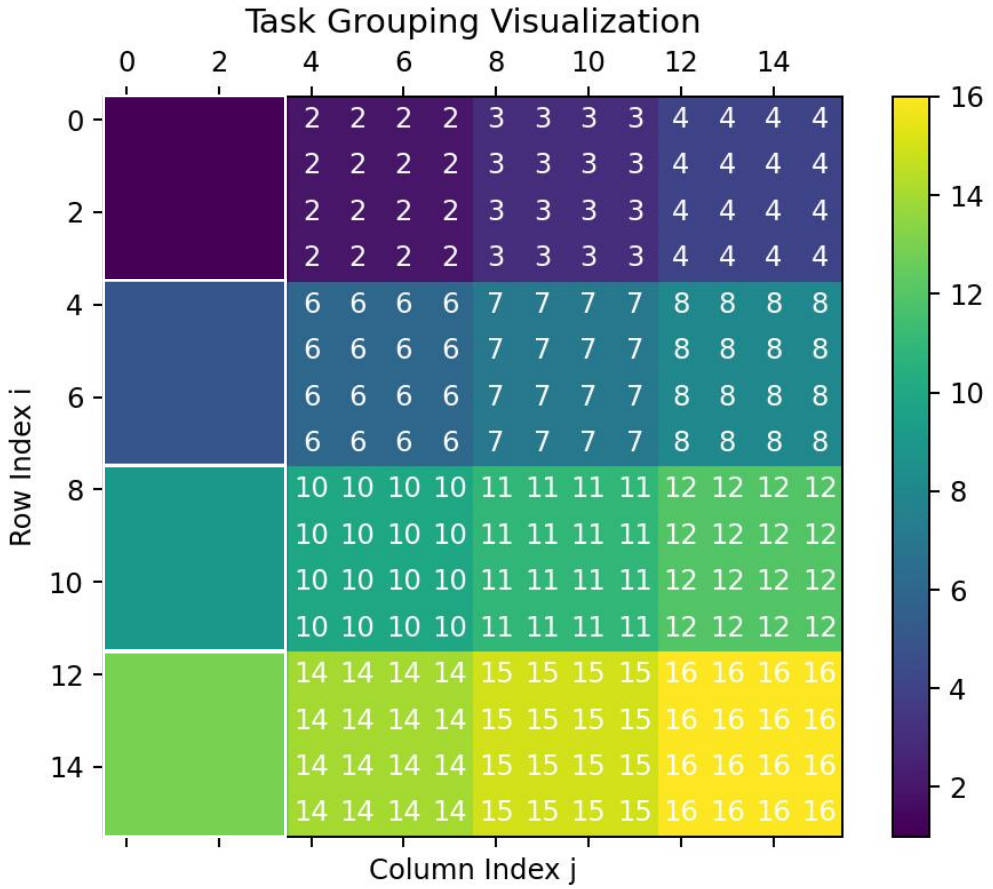
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



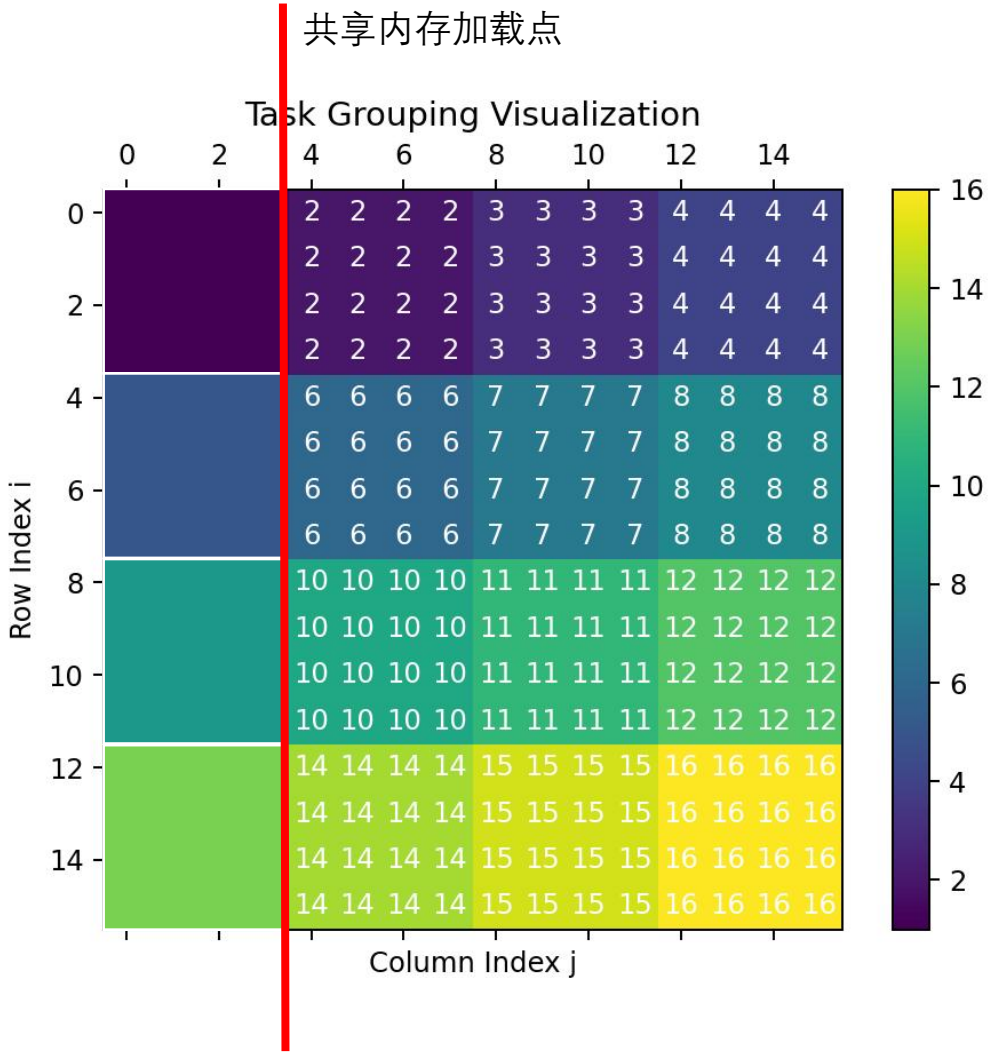
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



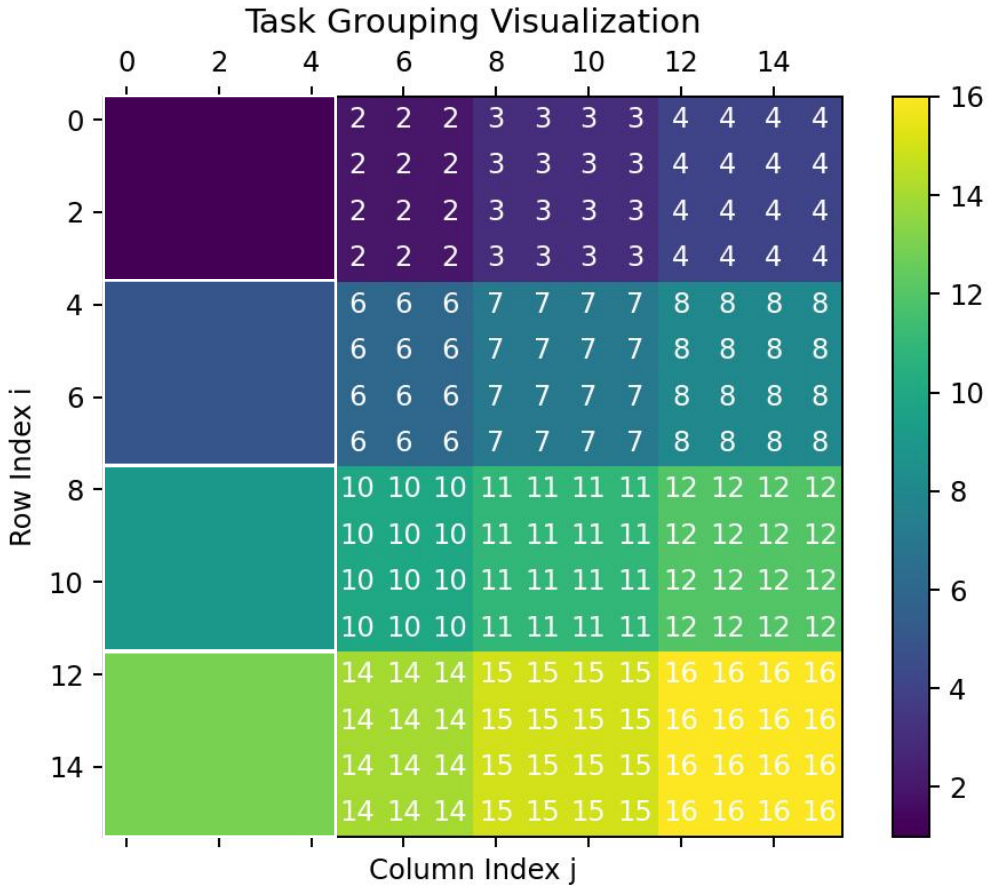
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



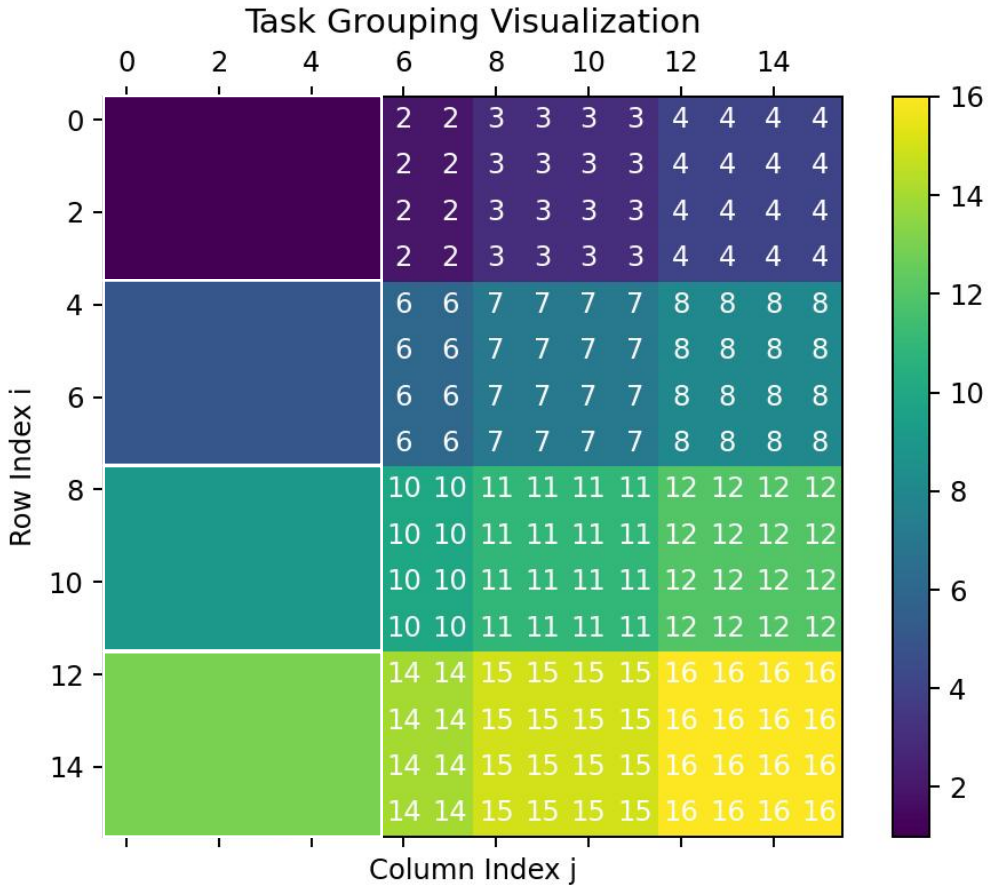
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



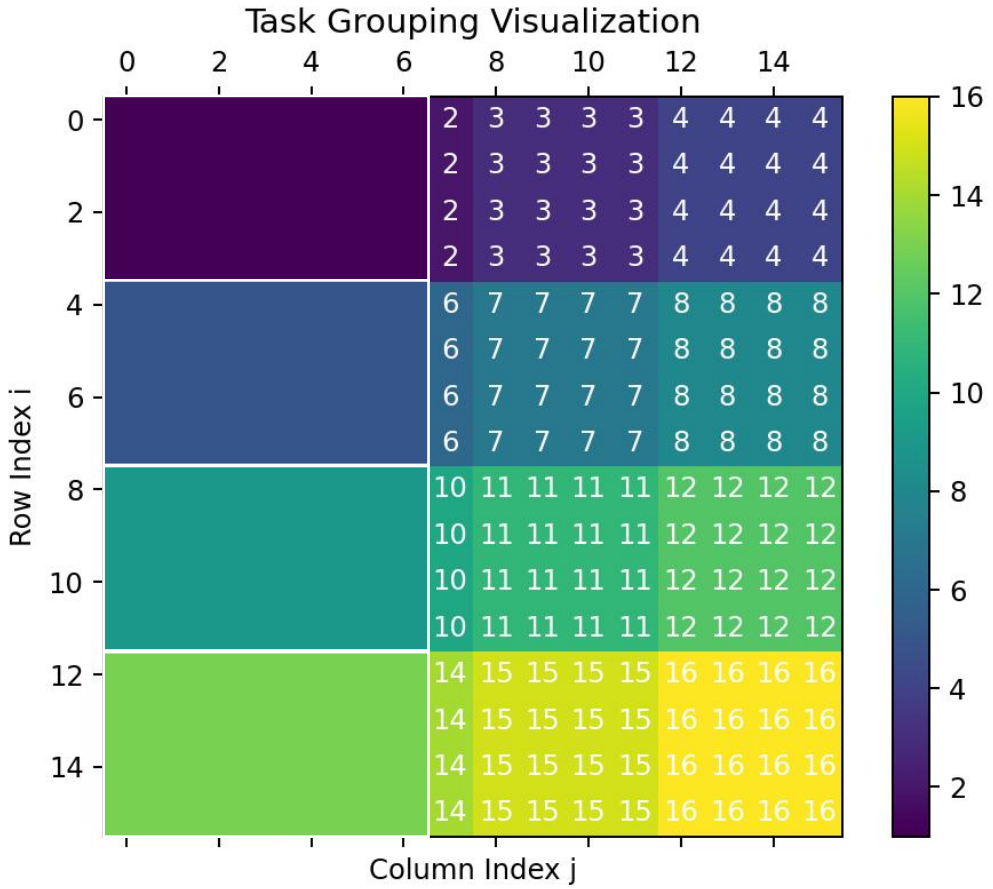
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



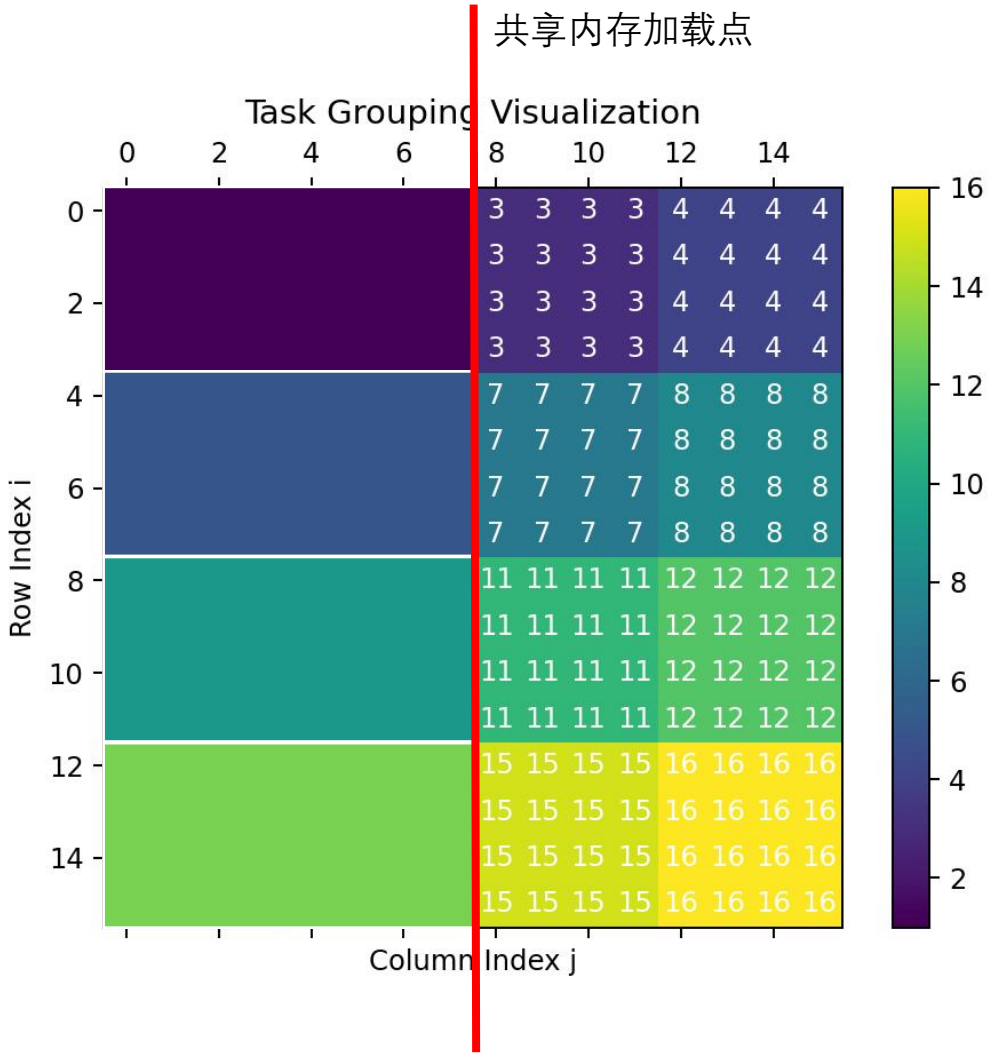
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



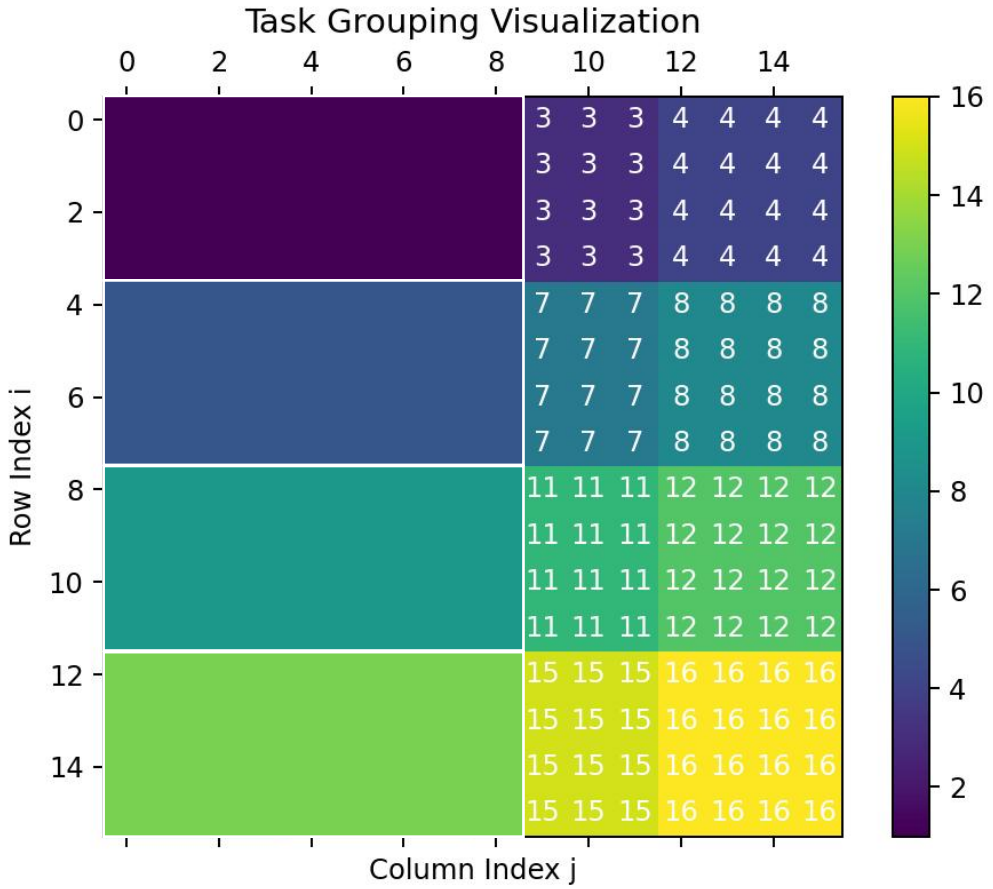
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



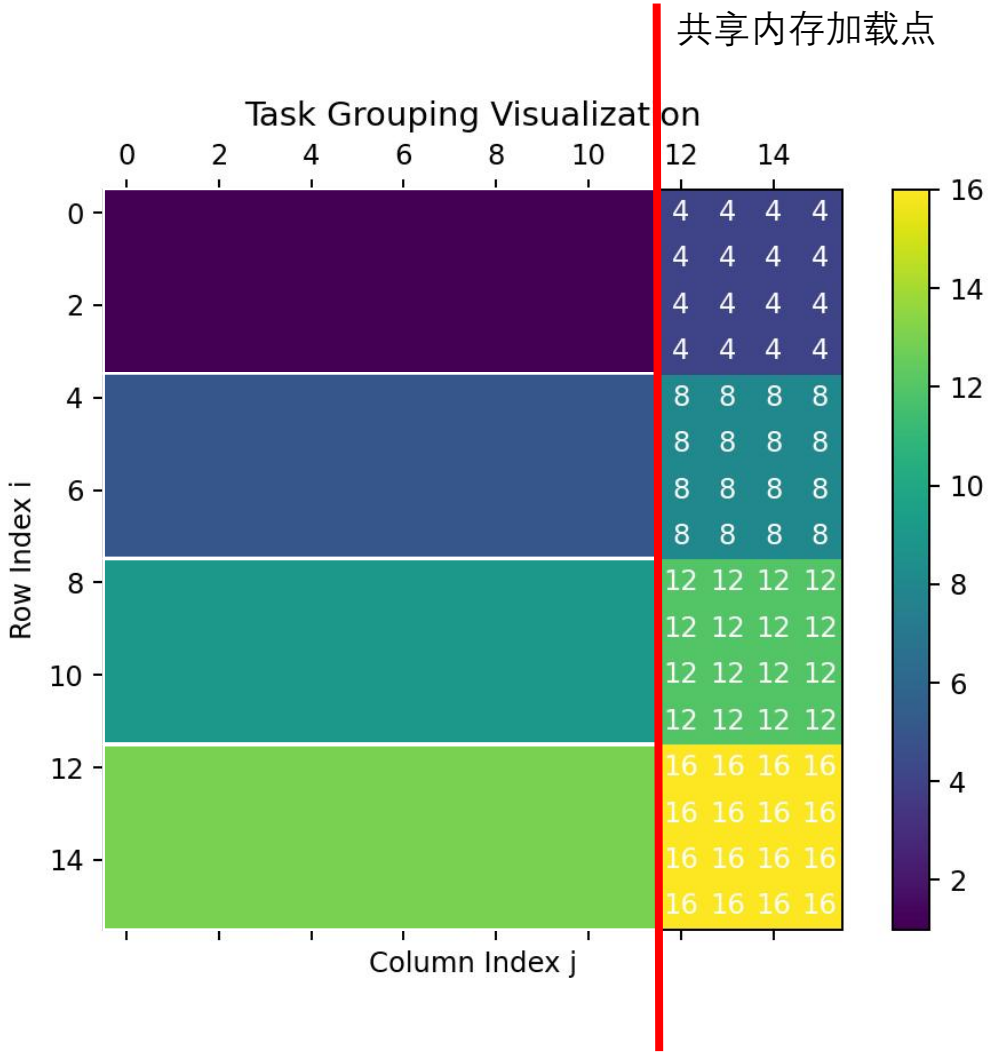
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



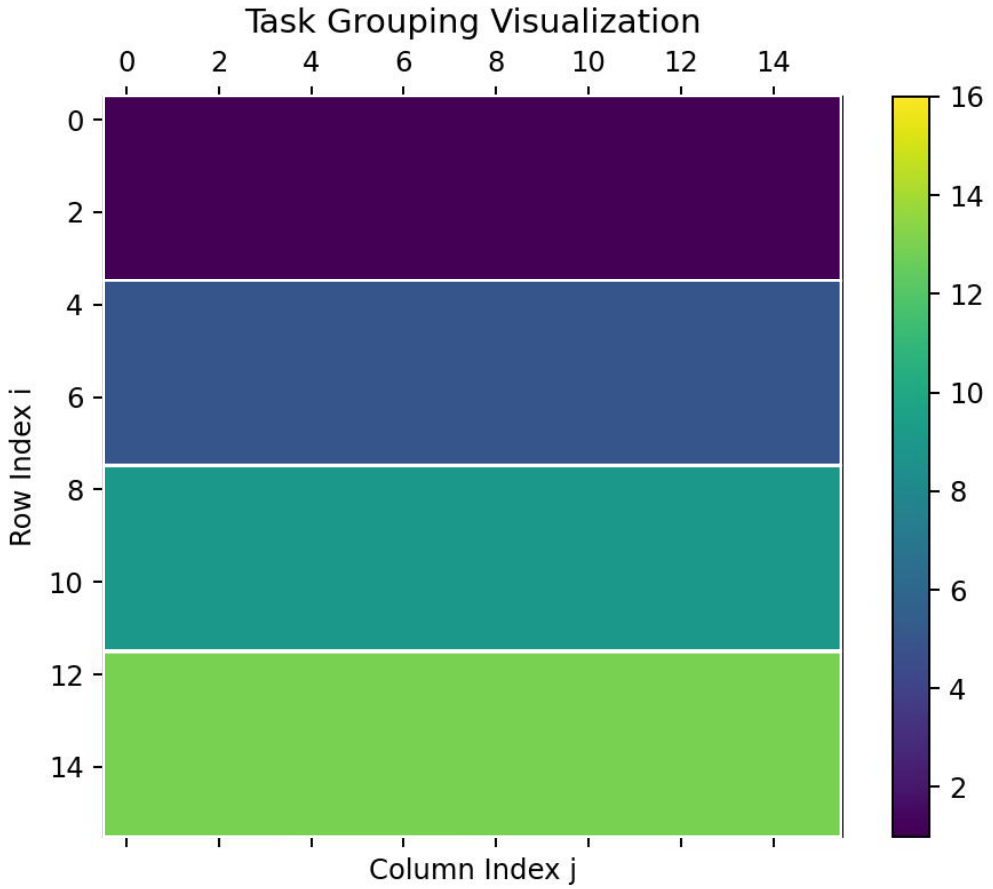
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



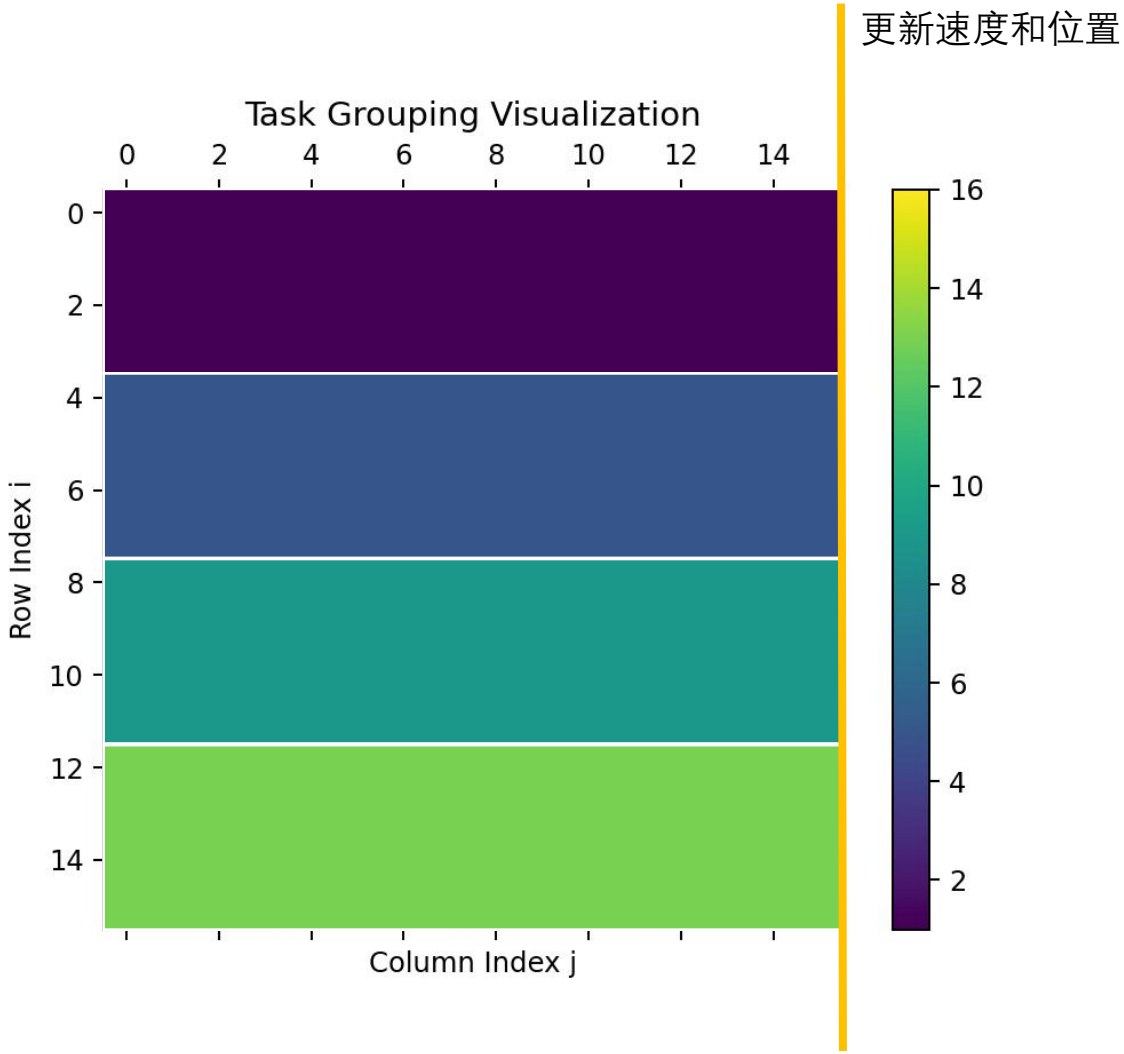
[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



[02] 方法B

1.直接计算各个天体的加速度

- 对于1个天体。其受到所有n个天体的作用产生加速度。（自己对自己的作用记为0）
- 直接计算加速度，并直接加和。
- 一个线程对应一个物体，加速度在局部变量中累积。

2. 更新位置

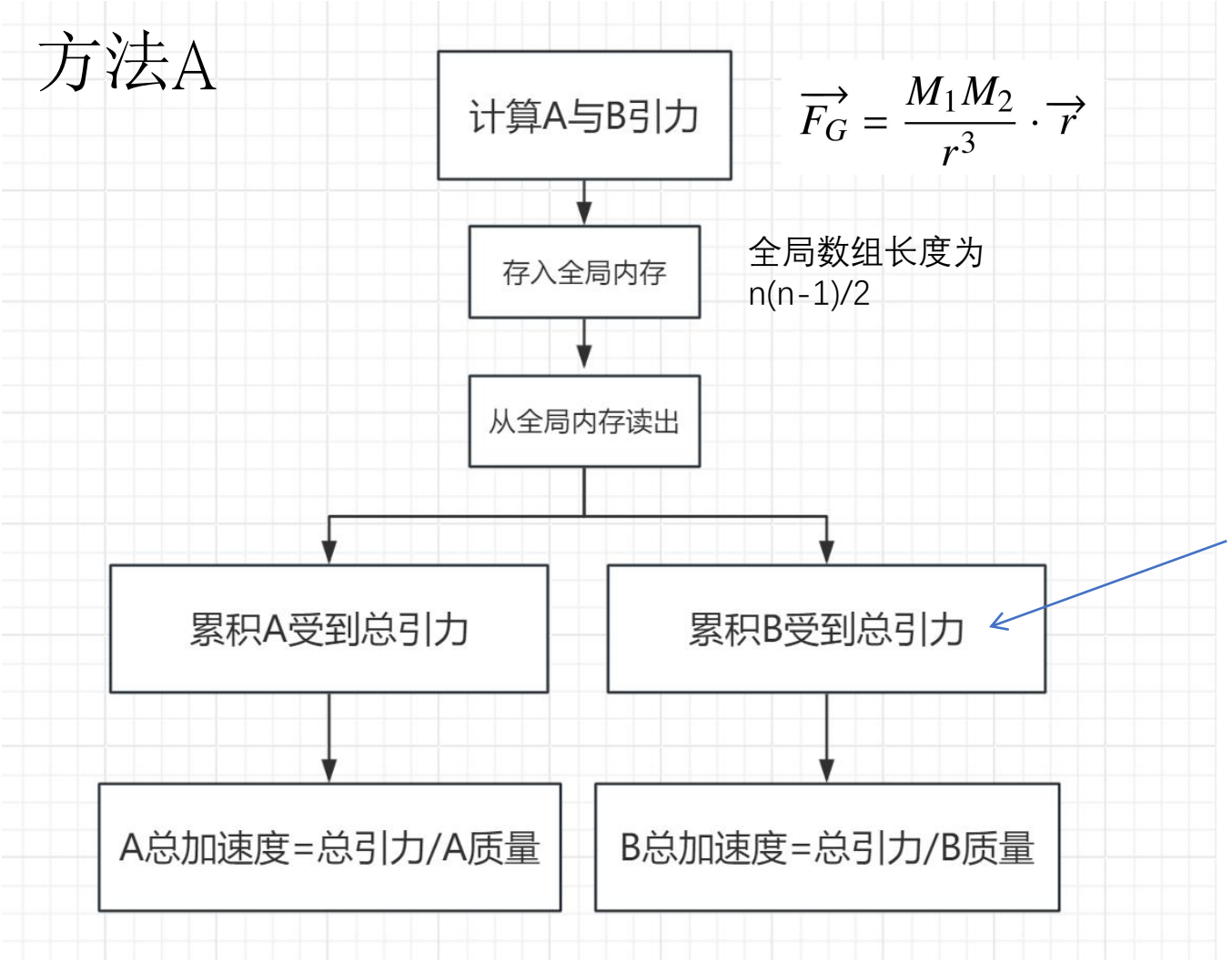
- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

4ms

计算量更大，
但是反而更快？？

[02] 思考

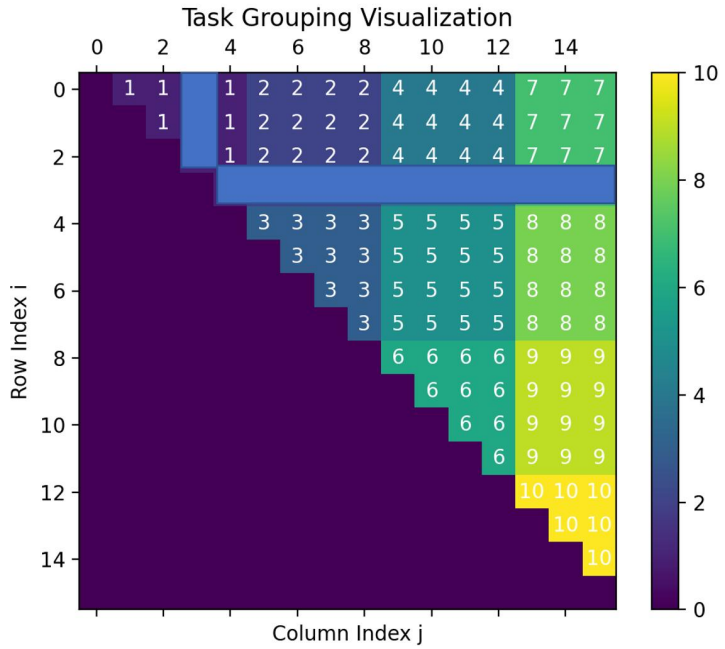
方法A



$$\overrightarrow{F_G} = \frac{M_1 M_2}{r^3} \cdot \overrightarrow{r}$$

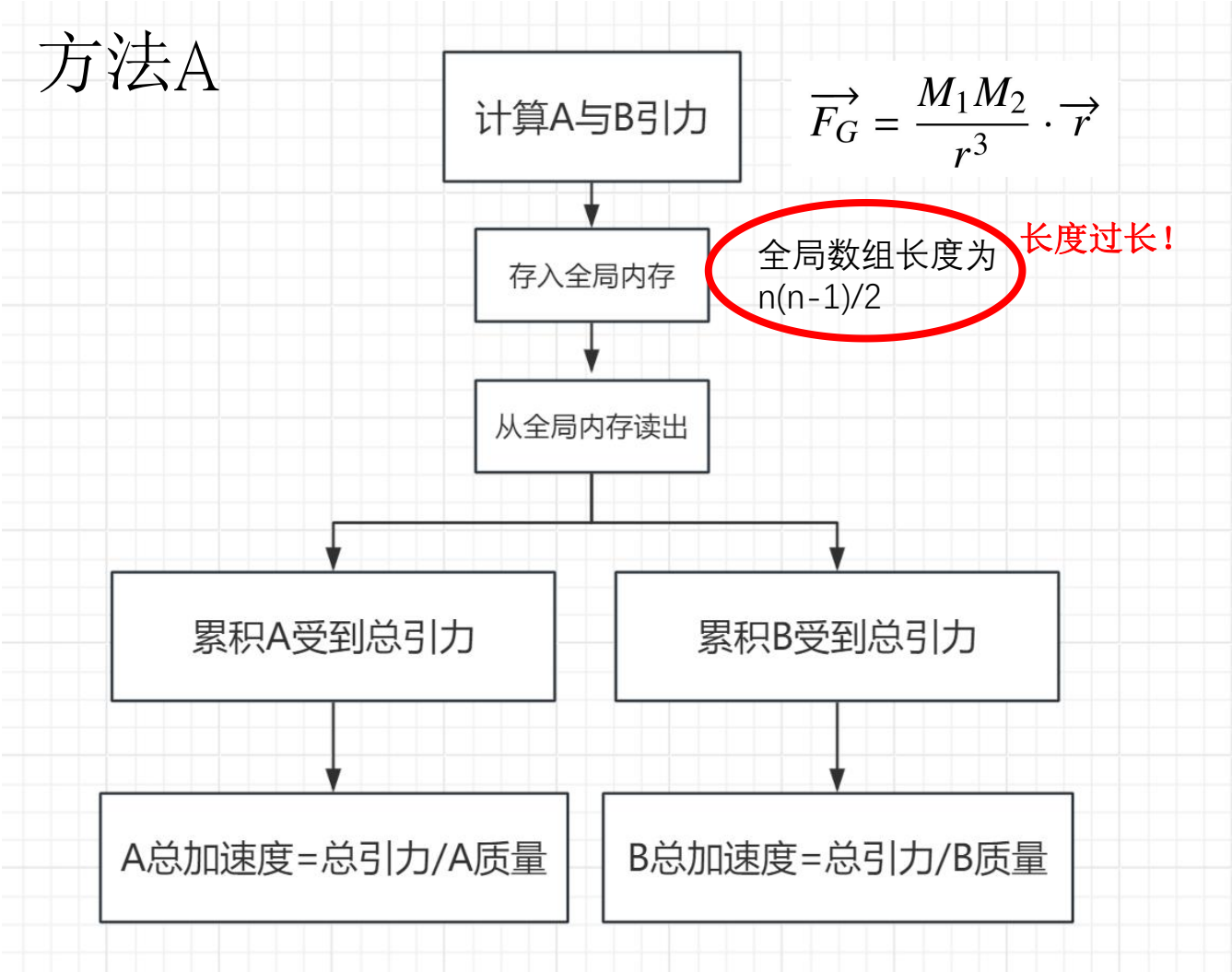
全局数组长度为
n(n-1)/2

计算天体3收到的总引力G[3]



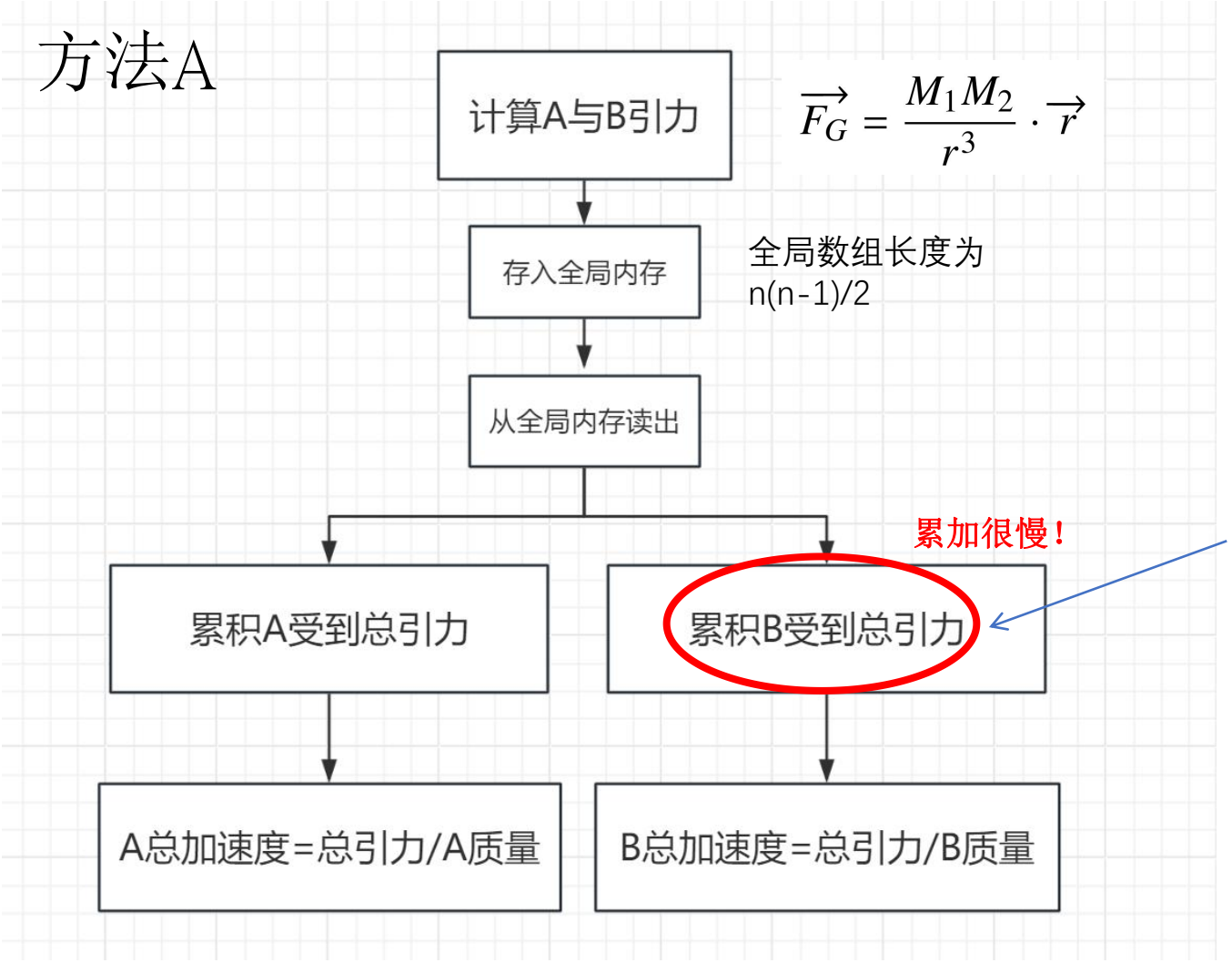
[02] 思考分析

方法A

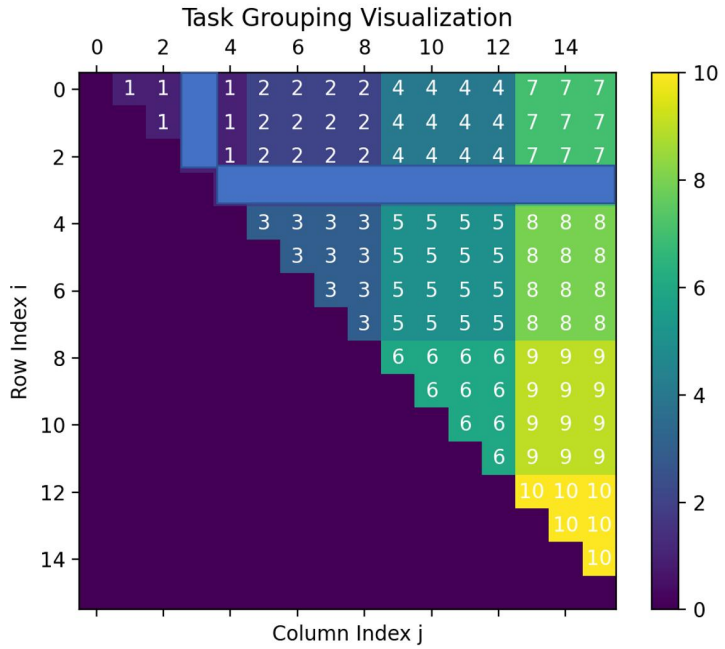

计算A与B引力
$\overrightarrow{F_G} = \frac{M_1 M_2}{r^3} \cdot \overrightarrow{r}$
存入全局内存
全局数组长度为
n(n-1)/2
长度过长！
从全局内存读出
累积A受到总引力
累积B受到总引力
A总加速度=总引力/A质量
B总加速度=总引力/B质量

[02] 思考分析

方法A

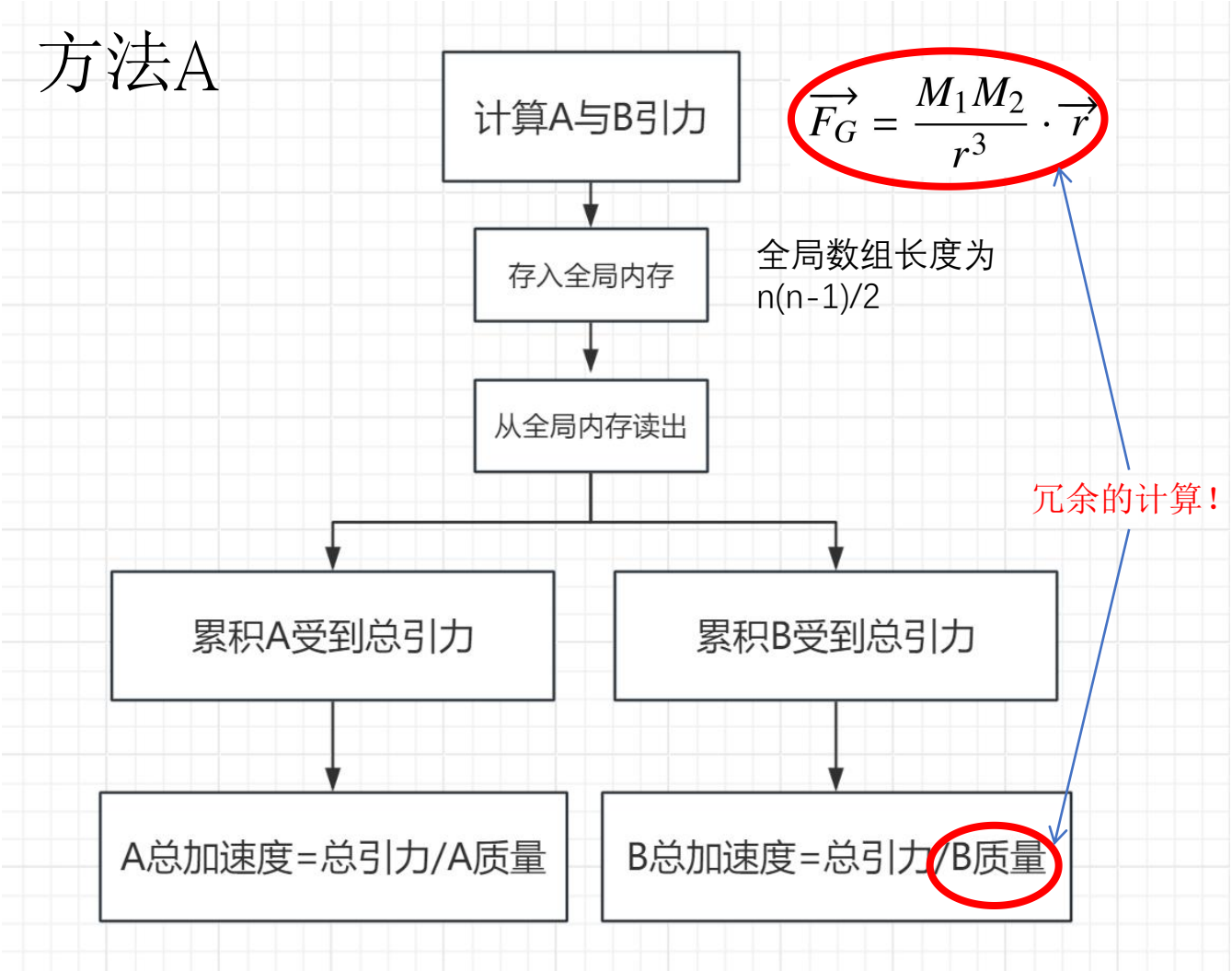


计算天体3收到的总引力G[3]



[02] 思考分析

方法A



$$\overrightarrow{A\text{受到来自}B\text{的加速度}} = \frac{G\frac{M_A M_B}{|AB|^2} \cdot \frac{\overrightarrow{AB}}{|AB|}}{M_A}$$

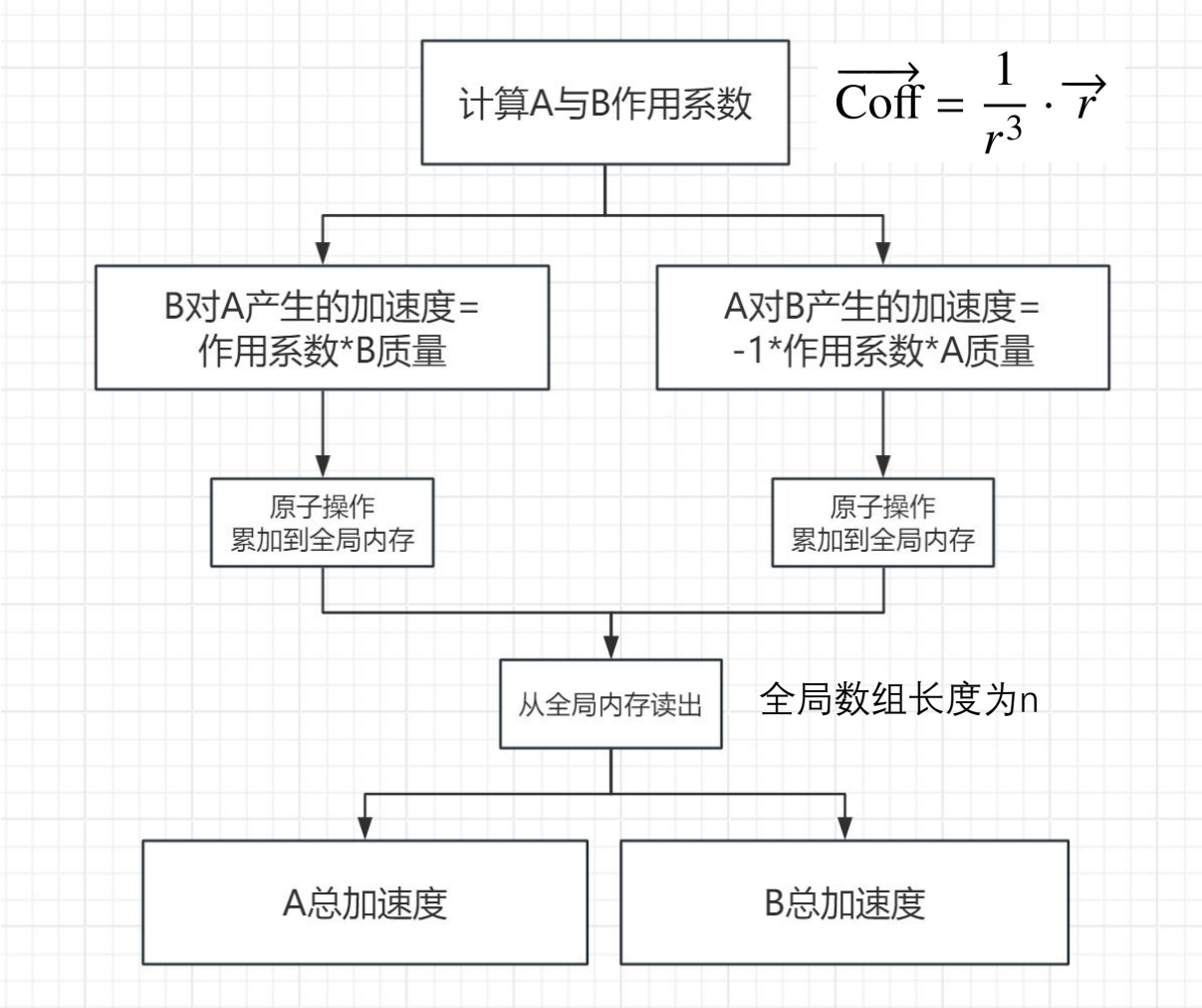
$$= \frac{\overrightarrow{AB}}{|AB|^3} M_B$$

$$\overrightarrow{B\text{受到来自}A\text{的加速度}} = \frac{G\frac{M_B M_A}{|BA|^2} \cdot \frac{\overrightarrow{BA}}{|BA|}}{M_B}$$

$$= \frac{\overrightarrow{BA}}{|BA|^3} \cdot M_A$$

$$= (-1)\frac{\overrightarrow{AB}}{|AB|^3} \cdot M_A$$

[02] 方法C


计算A与B作用系数
$\overrightarrow{Coff} = \frac{1}{r^3} \cdot \vec{r}$
B对A产生的加速度=
作用系数*B质量
A对B产生的加速度=
-1*作用系数*A质量
原子操作
累加到全局内存
原子操作
累加到全局内存
从全局内存读出
全局数组长度为n
A总加速度
B总加速度

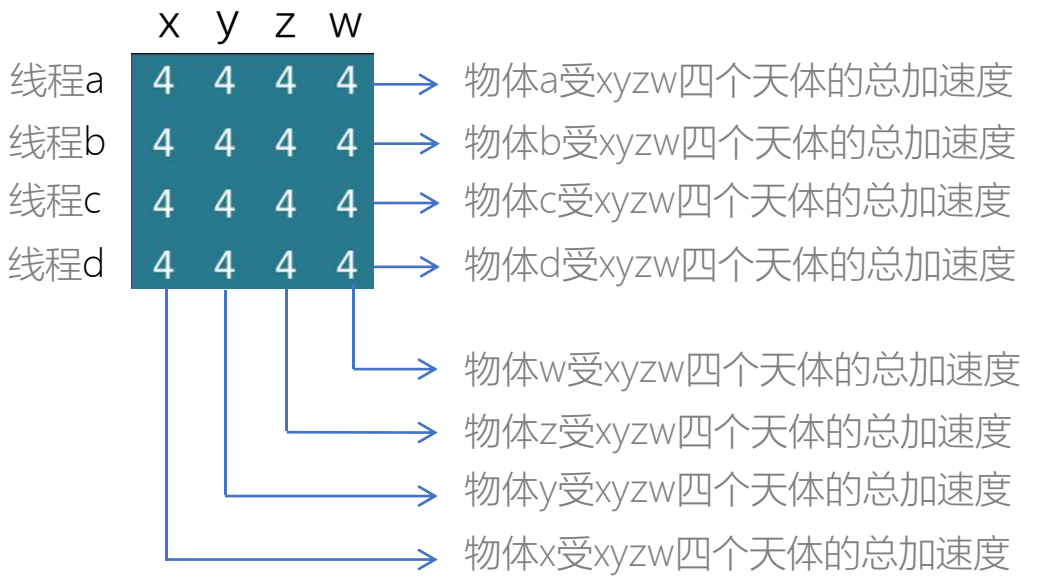
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

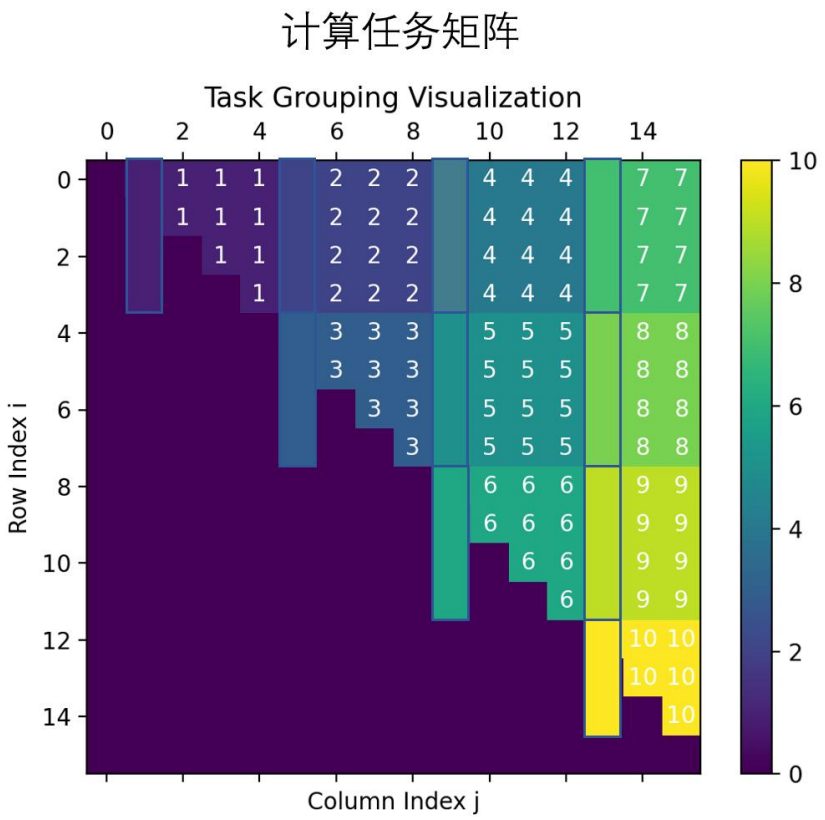
- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



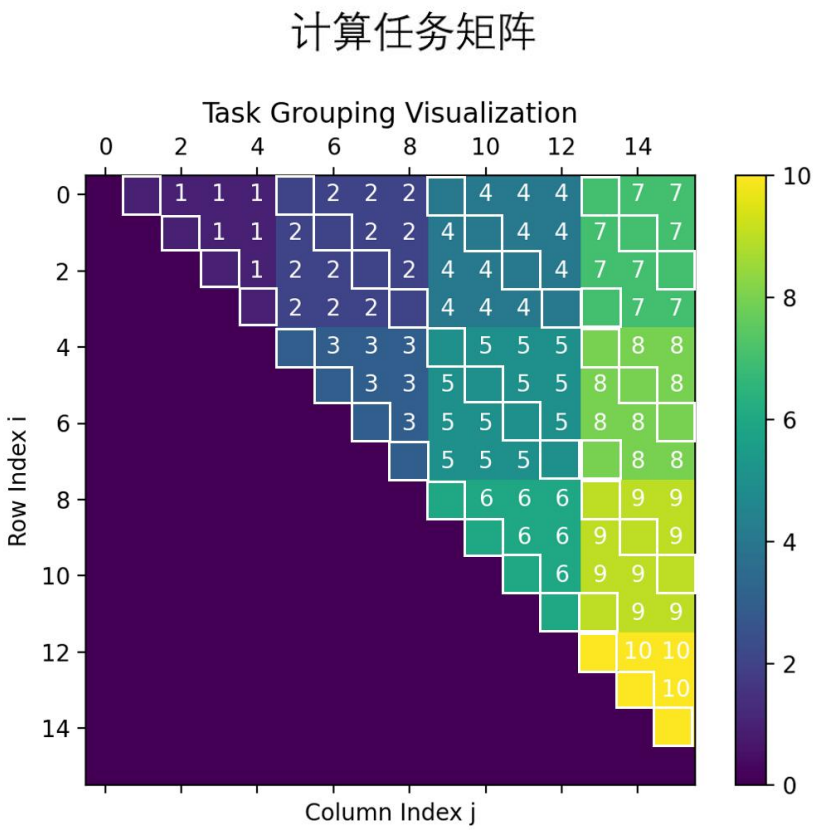
行和：直接保存在线程局部变量内

列和：保存在块内共享内存中，使用原子操作累积（冲突避免！）

[02] 方法C



原子操作
冲突避免！



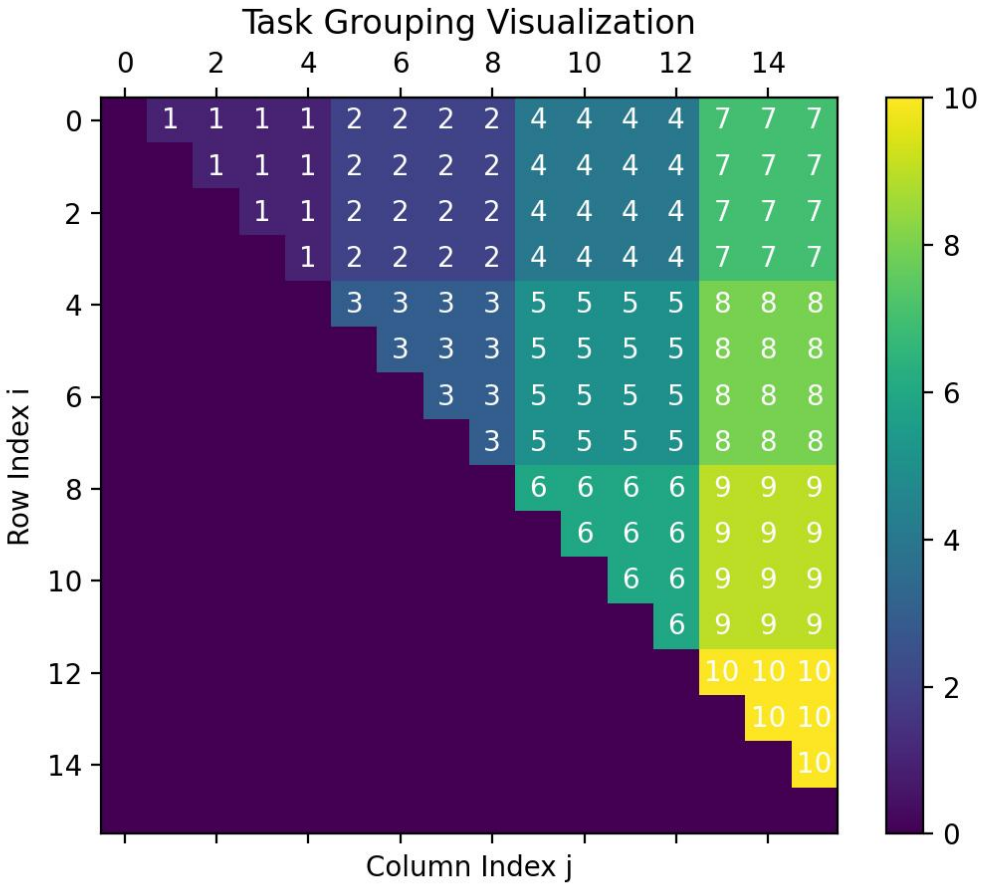
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



[02] 方法C

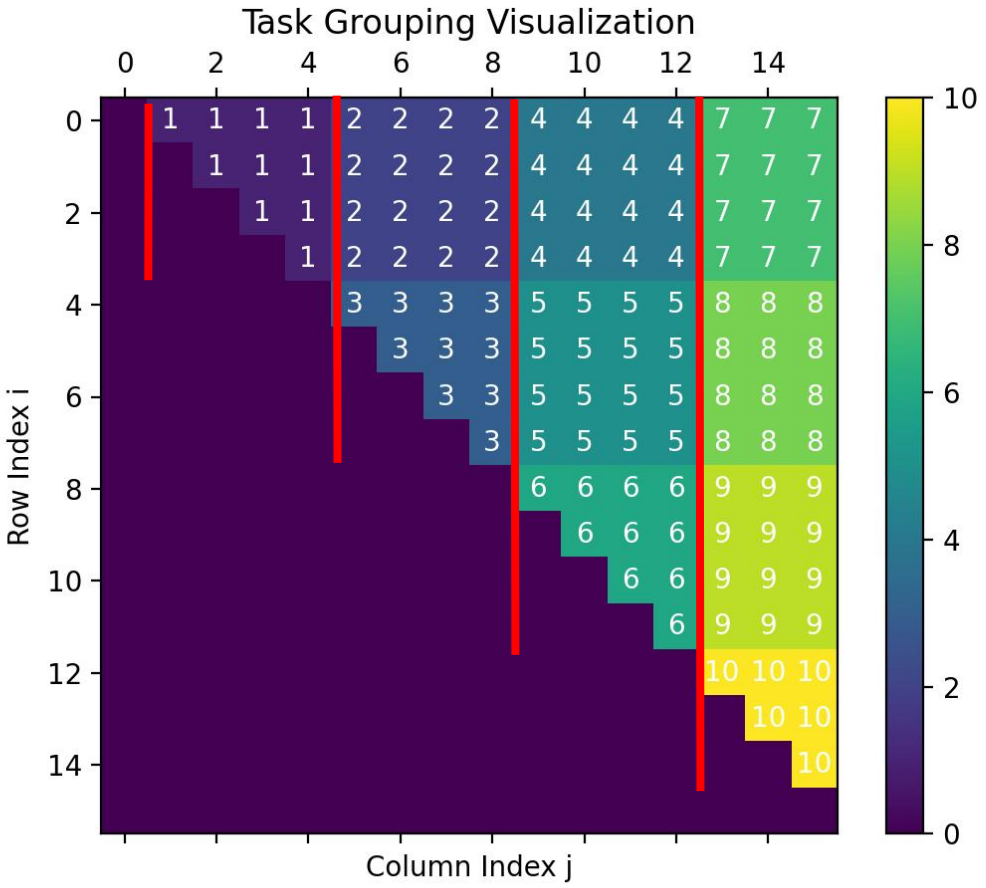
1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

共享内存加载



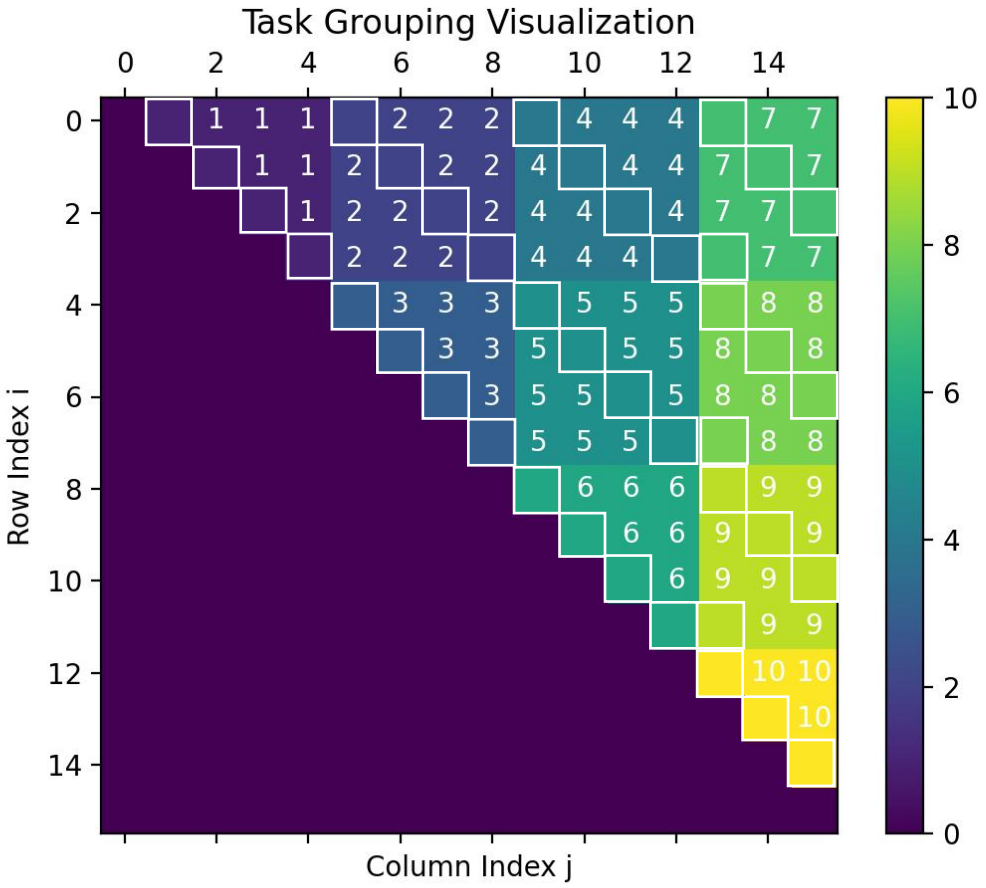
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



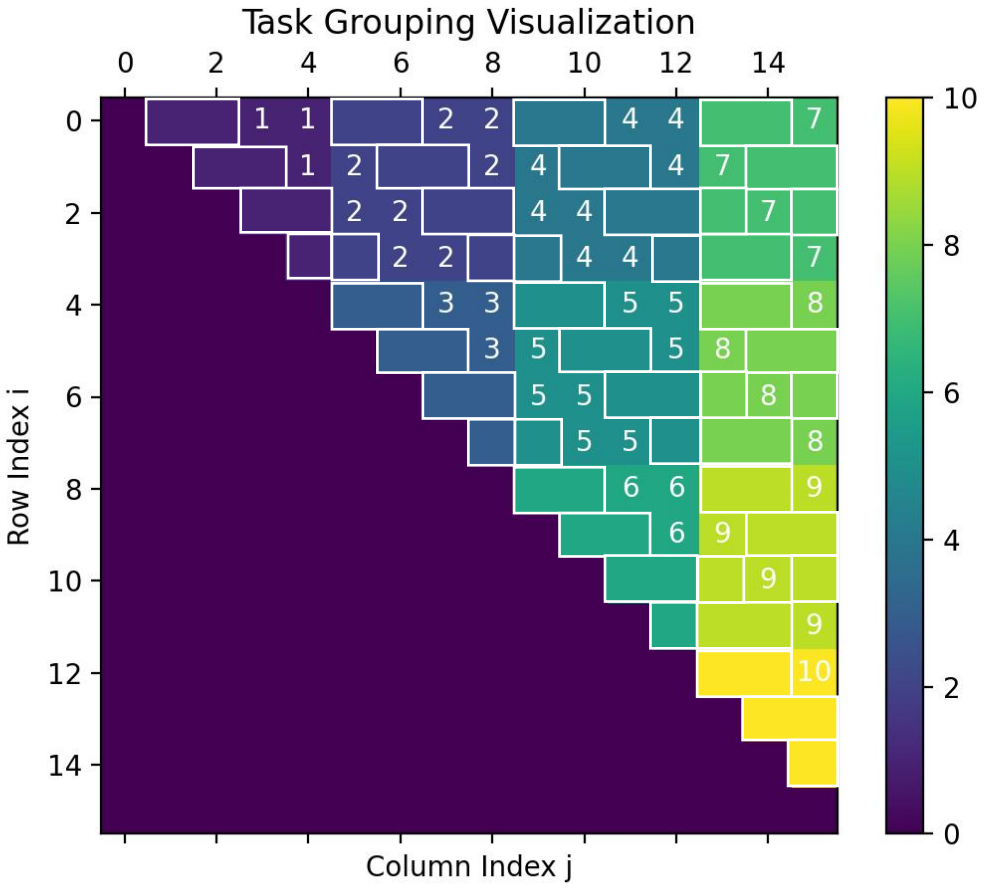
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



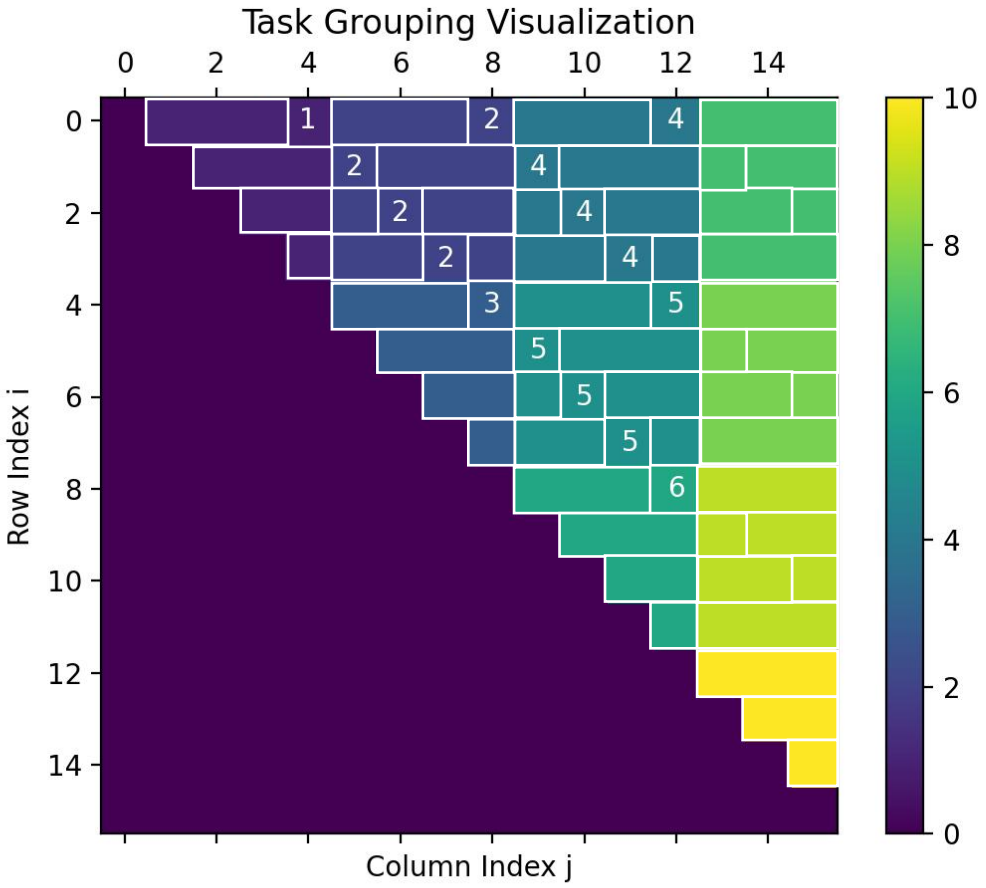
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



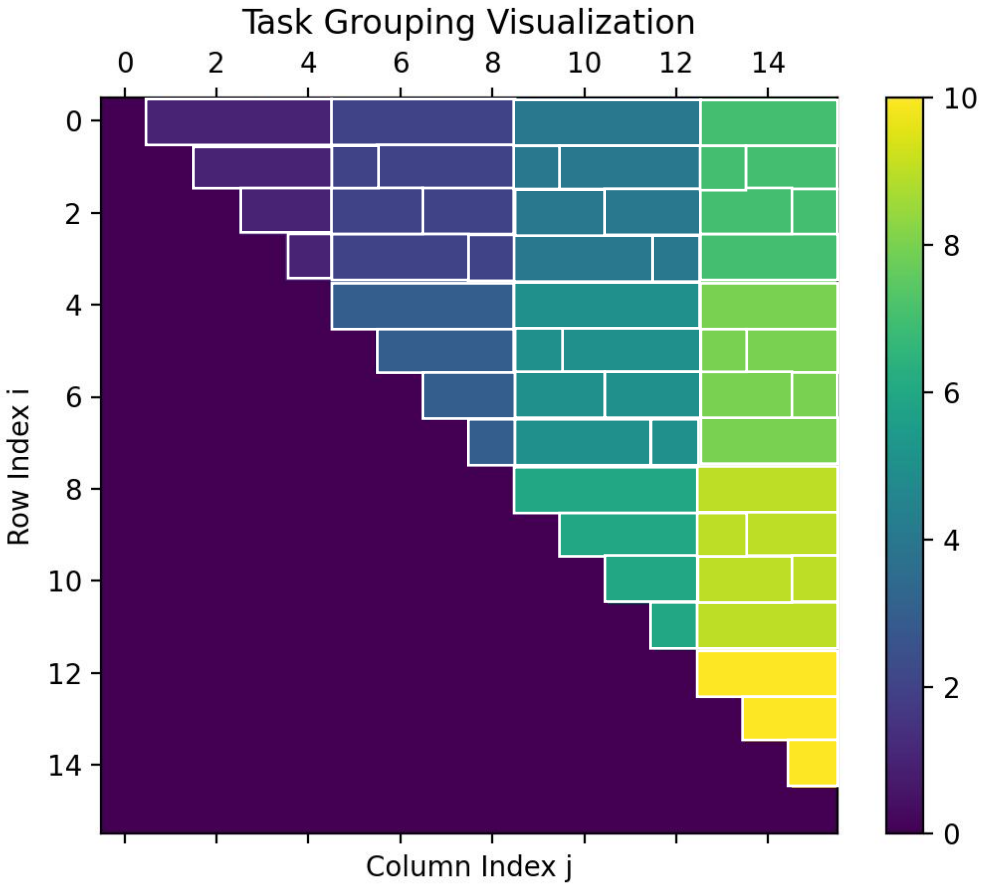
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



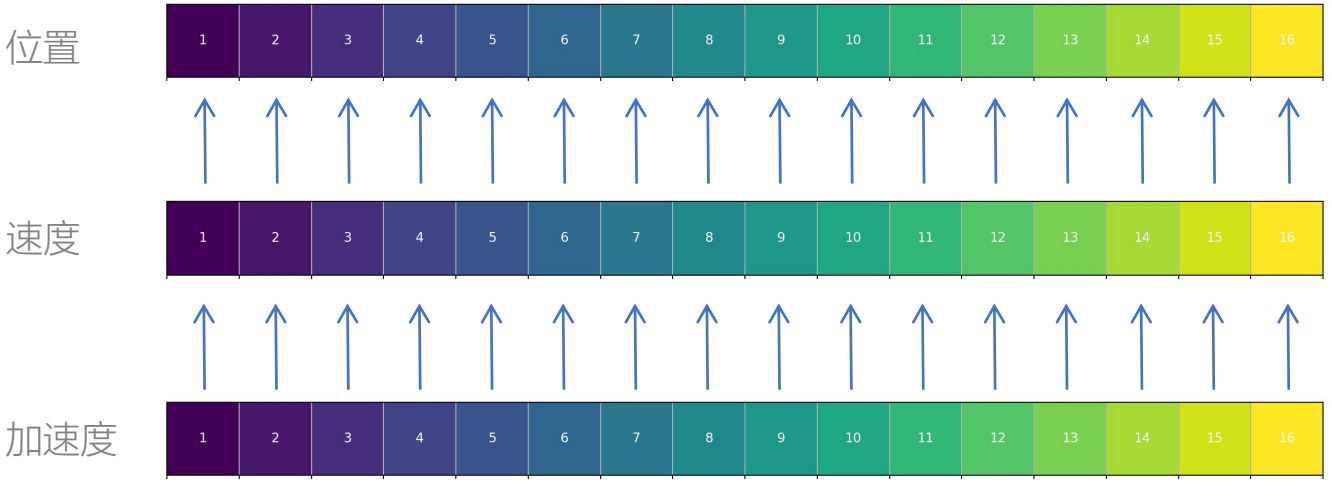
[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置



[02] 方法C

1.直接计算各个天体的加速度

- 对于N个天体，他们之间有N^2个相互作用力
- 使用万有引力定律计算相互之间的加速度
- 不计算重力直接计算加速度，并直接加和

1.5ms

惊人的提升!

2. 更新位置

- 确定一个最小时间刻
- 结合天体当前的速度和加速度更新天体速度
- 使用更新后的天体速度更新天体位置

0.3ms

[Part 03]

比较分析

[03] 方法C阶段I

```
// 运行时间1.5ms
// (N/BlockSize)*(N/BlockSize-1)/2*BlockSize个线程并行
__global__ void cal_acc_new(float3* acc_array, real* position_and_weight, int block_edge_len,
int hori_group_nums, int N) {
    extern __shared__ real sharedMemory[];
    <4次全局内存加载> // 将该线程负责的横向天体数据载入局部数据
    <4次全局内存加载> // 将一个其他天体的数据载入共享内存
    <一次同步>// 加载完毕后块内同步

    // BlockSize次循环
    for (int i = start_local_index;i < block_edge_len;i++) {
        <30次float乘法运算+6次浮点加法运算> // 计算加速度
        <3次float加法运算>  // 累积行和
        <3次对于共享内存的原子操作> // 将计算值加入到'列和数组'中
    }
    <一次同步> // 计算完所有块内节点后块内各线程同步

    <3次对于全局内存的原子写操作> // 保存行和到全局内存
    <3次对于全局内存的原子写操作> // 保存列和到全局内存
}
```

[03] 方法C阶段II

```
// 运行时间0.3ms
// N个线程并行
__global__ void use_acc_update_position(real* position_and_weight, real* m_deviceVelocity,
float3* acc_array) {
    <1次全局内存访问> // 从全局内存中取出加速度数据
    <3次全局内存访问> // 从全局内存的速度数组中取出三个方向的速度数据

    <3次float乘法运算> // 计算新的速度
    <3次全局内存访问> // 将更新后的速度写回全局内存

    <3次float乘法运算>  // 计算新的位置
    <6次全局内存读写> // 将更新后的位置写入全局内存
}
```

[03] 方法B

```
// 运行时间4ms
// N个线程并行
__global__ void simple_update_all( real* m_deviceVelocity, real* position_and_weight, int N) {
    extern __shared__ real mass[];
    <4次全局内存访问>
    // N次循环
    while (now_pos < 4 * N) {
        // 一个线程加载一个数据，然后计算BLOCK_SIZE个，然后同步一下，然后再加载数据到共享内存，然后再同步一下
        <4次全局内存访问>
        <一次同步>
        // BlockSize次循环
        for (int i = 0;i < BLOCK_SIZE;i++) {
           <30次浮点乘法运算+6次float加法运算>
        }
        <一次同步>
    }

    <3次全局内存访问> // 从全局内存的速度数组中取出三个方向的速度数据

    <3次float乘法运算> // 计算新的速度
    <3次全局内存访问> // 将更新后的速度写回全局内存

    <3次float乘法运算>  // 计算新的位置
    <6次全局内存读写> // 将更新后的位置写入全局内存
    }
}
```

[03] 比较分析

	方法C阶段I	方法C阶段II	方法B
并行线程数	$N * (\frac{N}{BlockSize} - 1) * \frac{1}{2}$	N	N
全局内存访问（每线程）	8次	13次	4N+1次
浮点乘法运算（每线程）	$BlockSize * 30$次	6次	30*N+6次
同步（每线程）	2次	\	N/BlockSize次
共享内存原子操作（每线程）	$BlockSize * 3$次	\	\
全局内存原子操作（每线程）	6次	\	\

[03] 比较分析

带入数值
N=20480， Block Size=256

	方法C阶段I	方法C阶段II	方法B
并行线程数	80,896	20,480	20,480
全局内存访问（每线程）	8次	13次	81,921次
浮点乘法运算（每线程）	7,680次	6次	614,406次
同步（每线程）	2次	\	80次
共享内存原子操作（每线程）	768次	\	\
全局内存原子操作（每线程）	6次	\	\
实测耗时	1.5ms	0.3ms	4.0ms

[Part 04]

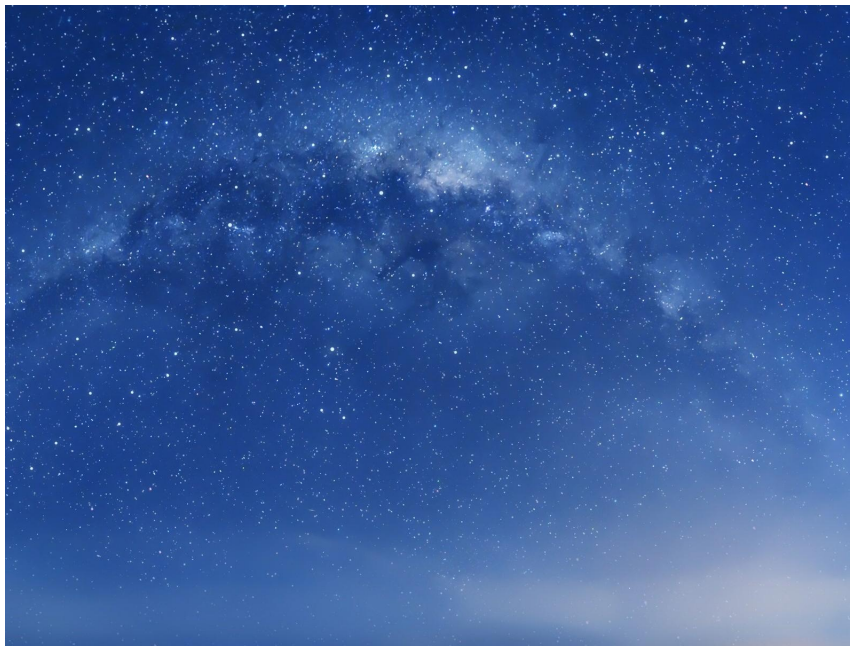
总结

[04]

计算任务量绝非唯一！

- 考虑内存访问和原子操作等开销
- 避免原子操作冲突很重要
- 简单的方法不一定就不好





感谢观看
下次再会

在微微的晚风下欣赏夜空

寻一个惬意的夜晚
看一次久违的满天星光
重温儿时搬个小板凳
在那片夜空下仰望天空的美好回忆