下面的实验测量了最经典的几种格式,COO, CSR, CSC在经典的sparse运算$SpMV, SpDMM, SpGeMM$中的开销。

过程中的所有代码都在zip文件中。

Experiment Setup

- Nvidia A100
- cuSPARSE in CUDA Toolkit 11.2

Test method

均匀地生成元素$x_{i,j} \in U[1,n] \times U[1,m]$

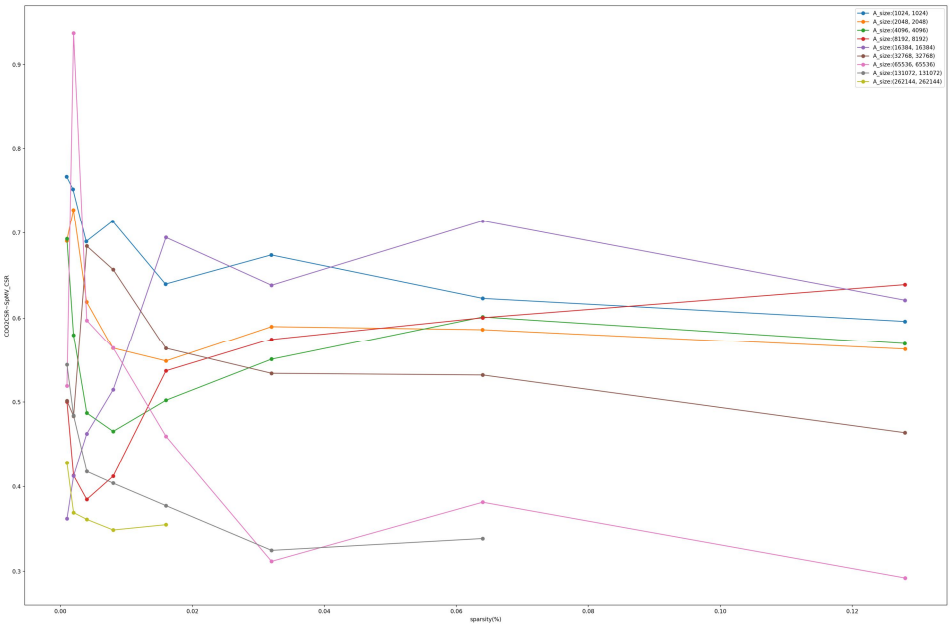
使用$cudaEvent$来测量$kernel$的运行时间。

对于每个$kernel$，进行10次运算，计算RSD(relative standard deviation)

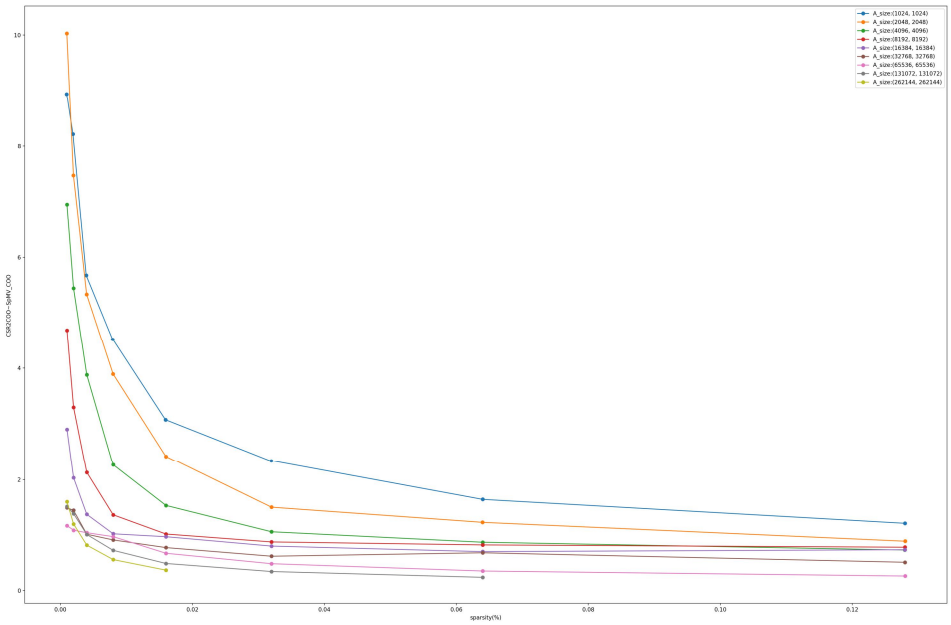
$$RSD = \frac{\sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})}{n-1}}}{\overline{x}} \times 100$$

Some results

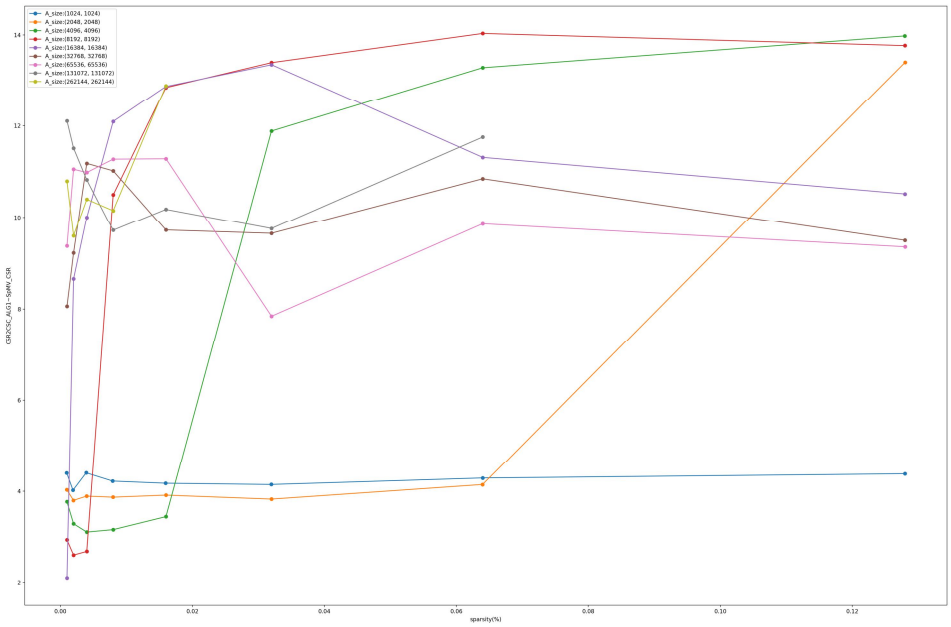
COO2CSR/SpMV(CSR)



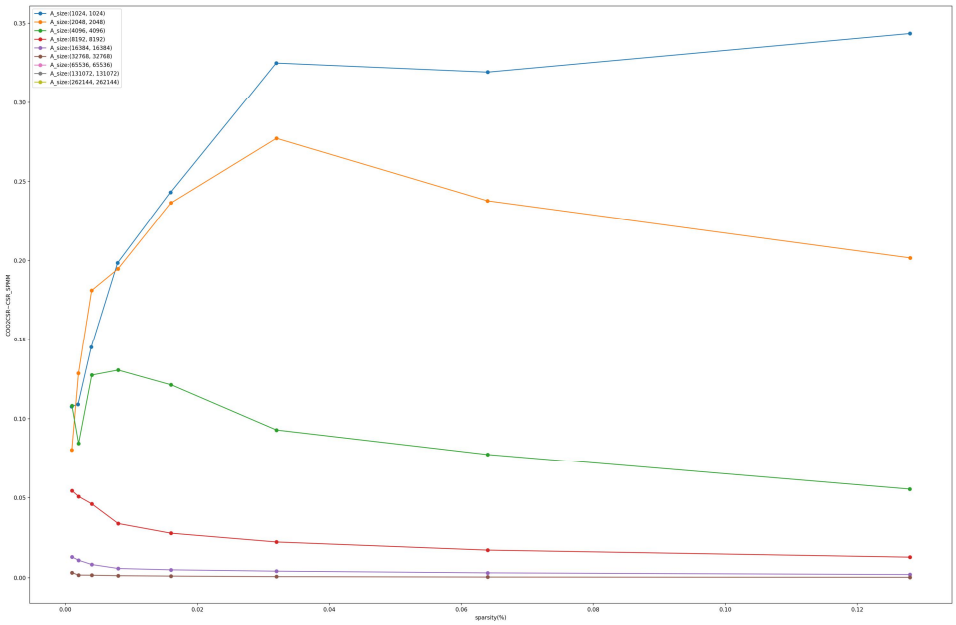
CSR2COO/SpMv(COO)



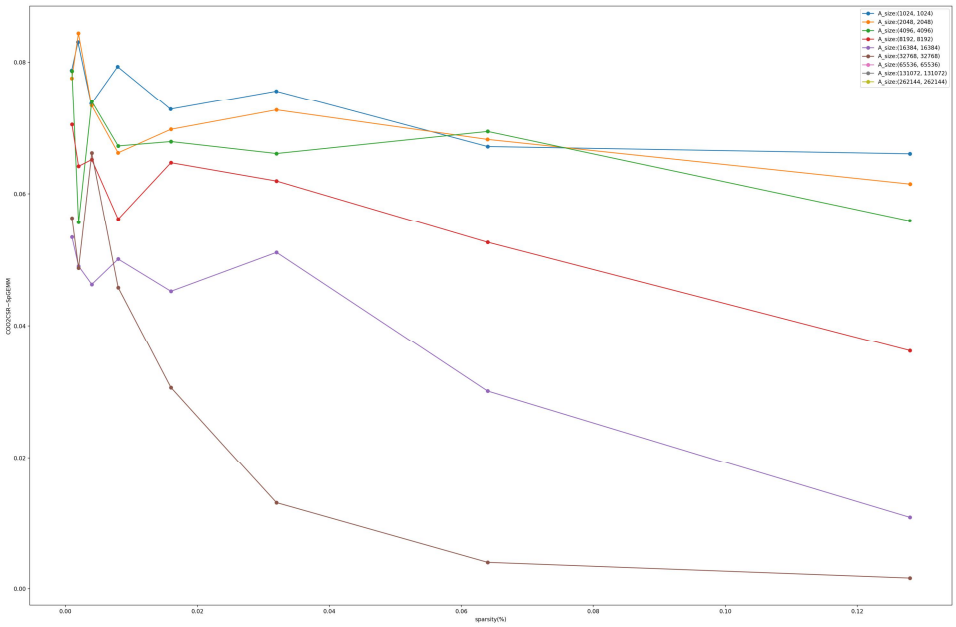
CSC2CSR/SpMV(CSR)



COO2CSR/SpMM(CSR)



COO2CSR/SpGEMM(CSR)



Conclusions

- COO2CSR是最快的格式转换，因为它只需要做一次前缀和，复杂度为$O(nnz/p + \log p)$
- 但是CSR2COO却显著慢于COO2CSR, 我个人认为原因主要是函数没有办法利用额外的缓存，所以很多操作都需要inplace进行，并且有可能函数并没有假设它的行是已经排好序的，所以导致性能显著慢于COO2CSR.
- CSR2CSC特别慢，甚至比CSR2COO然后COO2CSR还要来得慢。
- 格式转换的开销是无法忽略的，在SpMV里面， 根据不同的情况会有20%到200%的开销，在SpMM里面，会有5%到200%的开销，即使是在SpGeMM里面，最坏情况下格式转换的开销也能

和运算的开销持平。