

高性能计算实践

24秋 陈浩民 2023311426

实验环境：

OS版本 : 6.5.0 – 35 – generic
gcc版本 : (Ubuntu 11.4.0 – 1ubuntu1 22.04) 11.4.0
CPU型号 : 13th Gen Intel(R) Core(TM) i9 – 13900HX
CPU最大频率(MHz) : 2200.0000
CPU最小频率(MHz) : 800.0000
物理核数 : 24
内存(M) : 16095080(可用 : 9297836)

test_cblas_dgemm.c修改为行主序后结果有什么不同：

结果与先前完全不同，原因应该是，行主序和列主序交换过后，相当于矩阵转秩了，但是

$$(AB)^T = B^T A^T \neq A^T B^T$$

所以前后结果不同。

比较：

	256	1024	4096	8192
<i>cblas_dgemm duration</i>	0.000996s	0.052410s	1.104756s	7.206882s
<i>naive_dgemm duration</i>	0.153926s	7.731068s	1364.368622s	≈ 9h
<i>cblas_dgemm gflops</i>	68.079601	81.949386	248.813228	305.128245
<i>naive_dgemm gflops</i>	0.436814	0.555546	0.201469	≈ 0.3

朴素矩阵乘法用时的增长速率很快，大概是矩阵阶数的三次方。而用*cblas*的算法，前期增长率很快，但是当矩阵的规模很大的时候用时增长率降低。*cblas*算法的浮点操作值随矩阵的规模的增大而缓慢增加，朴素乘法则基本不变且远小于*cblas*算法的值。原因是*cblas*的算法利用并行计算使得每秒能够进行操作的次数大大增多，而朴素算法一直采用同一种串行方式，每秒进行的操作次数稳定且较少。

碰到的问题：

在比较朴素乘法和并行计算时，矩阵的规模达到4096 × 4096后朴素乘法用时达到20多分钟，估算发现如果矩阵规模增加到8192 × 8192，计算时间将达到9小时，所以我采用了估算值。