

高性能计算实验报告

实验2：Linux环境下的C语言编程与矩阵乘法

2024年秋季学期 姓名：曹馨尹 学号：2023311708

一、实验环境

OS版本：Ubuntu 24.04 LTS
GCC版本：13.2.0
CPU型号：AMD Ryzen 9 7940H w/ Radeon 780M Graphics
CPU频率：7984.98 MHz
CPU物理核数：2核
内存：
主内存 (Mem)
总内存 (total): 3,993,728 KB (约 3.99 GB)
已用内存 (used): 1,646,976 KB (约 1.65 GB)
空闲内存 (free): 1,634,580 KB (约 1.63 GB)
共享内存 (shared): 62,704 KB (约 62.7 MB)
缓冲/缓存内存 (buff/cache): 1,003,820 KB (约 1.00 GB)
可用内存 (available): 2,343,932 KB (约 2.34 GB)
交换内存 (Swap)
总交换内存 (total): 399,596 KB (约 399.6 MB)
已用交换内存 (used): 432,784 KB (约 432.8 MB)
空闲交换内存 (free): 356,812 KB (约 356.8 MB)

二、test_cblas_dgemm.c 修改为行主序后结果有什么不同？

首先，结果是相同的。但数据存储方式不同，性能也可能不同。

三、time_dgemm.c 分别测试M=N=K：256，1024，4096，8192时，下面表格的形式记录两者的duration和 gflops的值。从数据中可以发现什么规律，可以尝试自己分析下。数据有波动是正常的，可以运行多次取平均。

	256	1024	4096	8192
cblas_dgemm duration	0.002056	0.045232	2.292509	27.114001
naive_dgemm duration	0.058824	4.376031	641.285714	9128.363172
cblas_dgemm gflops	32.640498	94.954176	119.902651	81.102868

	256	1024	4096	8192
naive_dgemm gflops	1.140842	0.981476	0.428636	0.235735

分析：

- (1) duration:执行时间随矩阵大小增加而增加，与cblas相比，朴素算法的增速更大。
- (2) gflops：随矩阵大小增加而减小。

对比之下，naive_dgemm 是简单的三重循环实现，计算复杂度为 $O(n^3)$ ，缺乏优化，导致在处理大矩阵时性能急剧下降。

四、碰到的问题和解决办法

在往远程仓库推送文件时，我发现不能直接输入github用户名和账号，而是需要用个人访问令牌或SSH密钥。为了方便直接在github网站上手动上传。但是晚点还是去弄SSH密钥（令牌据说要每次推送手动输入，还是选择SSH）。