OpenMP 版本矩阵乘法实验报告

信息学院图灵实验班 常皓飞

针对矩阵乘法的优化问题，我采用了并行、矩阵转置、矩阵分块、内存对齐、编译优化等一系列优化方式之后，成功地在较大规模下相较无优化的做法提升了300倍左右的性能，**从约0.4 GFLOPS/s到约130 GFLOPS/s。**

问题描述

问题以矩阵相乘为背景，用程序实现起来并不难，难点是如何进行优化、更加高效地进行计算。

题目中要求利用 OpenMP 进行优化，但经过实验验证，我发现单纯这种策略所取得的优化程度不令人满意，与理论峰值性能相去甚远。因此，我针对这样一个经典的问题进行了更加深入的探索，并找到了一种更加高效的解决策略。

解题策略与过程

从25日，开始针对性的学习这一问题。到26日下午，基本得到现在的最终优化方案，并进行了验收。最终上传的五个版本的代码，从基础一步步迭代优化，正是自己的解题思路与过程。

`v0-basic.cc` 为提供的基准程序（为了进行更大规模的数据测试，进行了 `malloc` 优化），后续均在此基础上进行优化。

首先进行的是最基础的并行优化（`v1-basic.cc`）。通过 `omp_set_num_threads(16)` 和 `#pragma omp parallel for` 在执行程序时充分利用线程优势并行计算。

如果不深入探究更深层次的问题，工作也只能到此为止。但网络上有不少针对 `GEEM` 的讨论，从中不难捕捉到一个关键词：内存优化。优化的方向有很多，首先一个比较容易实现的优化是通过矩阵转置进行的内存优化（`v2-transform.cc`）。我将其理解为，二维数组在内存中的存储模式是行优先，在进行大量计算的时候，如果一个矩阵需要跨行读取，那这样的模式会影响缓存命中率，大量无用数据被读取到缓存中。CPU拥有三级缓存架构，缓存容量有限但访问更快。CPU进行运算时，缓存命中率对于提升性能至关重要。能够一定时间内从缓存中读取更多有效数据，就意味着计算速度的提升。接着尝试了矩阵的分块（`v3-block.cc`）。通过矩阵的分块，我们可以更好地利用缓存，也将提高计算效率。

到这里，一些基本的优化思路已经实现了。但是我注意到，此时距离峰值性能依然相去甚远。因此，我尝试进行更高级的优化，将数据进行内存对齐（`v4-cache.cc`）。在最内层循环，我使用了 `SIMD` 进行优化。

一个比较值得注意的细节是：

```
    for (int i = 0; i < BLOCK_SIZE; i++) {
        // for (int j = 0; j < BLOCK_SIZE; j++) {
        for (int k = 0; k < BLOCK_SIZE; k++) {
            #pragma omp simd // SIMD optimization for inner loop
            // for (int k = 0; k < BLOCK_SIZE; k++) {
            for (int j = 0; j < BLOCK_SIZE; j++) {
                localC[i][j] += localA[i][k] * localB[k][j];
                // localC[i][j] += localA[i][k] * localB[j][k];
            }
        }
    }
```

注意这里的循环嵌套，从数学意义上来讲等价，但考虑到计算机的存储、二维数组行优先等等特性，保持内部计算时两个矩阵数据对齐，这样在最内层就能够提高命中率，加快访问速度，对CPU更加“友好”。

将这些优化方法集合起来，形成了最终的优化版本。

最后进行的、也是最明显的优化方式，莫过于 `-O3` 编译优化。在此前进行测试时，并没有进行编译优化。因此，当时虽然已经写出了全部版本的方案，但我已经一筹莫展：在这种情况下，最优方案是 `v2`，即仅进行矩阵转置的方案，最高达到了 6 GFLOPS/s，大约提升了 10 倍的性能。这一结果虽然可以接受，但却意味着我所进行的分块、对齐等都毫无作用、甚至在拖后腿，让人不得不怀疑自己的思路与实现的正确性。但当我重新审视题目，使用 `-O3` 优化时，就变得豁然开朗。这带来了超过 10 倍的性能提升，让我的最终方案脱颖而出、结果一骑绝尘，证明了自己的努力与方向没有错误。

理论峰值性能计算

关于CPU理论峰值性能的计算，网络上似乎并没有统一的说法。比如说用核心数还是线程数，也有人表示这种计算毫无意义，实际性能会受到多种因素影响,如内存带宽、热量限制、实际工作负载等。我采用了一种比较广泛的说法，正确性有待验证：

$$\text{理论峰值性能} = \text{线程数} \times \text{单核主频} \times CPU\text{单个周期浮点计算值}$$

经过官网查询，本机CPU参数规格为12核心16线程，主频2.6GHz，支持AVX2指令集（每个时钟周期可以进行32次单精度浮点运算）。因此

$$\text{理论峰值性能} = 16 \times 2.6 \times 32 = 1,331.2 \text{ GFLOPS}$$

实验

26日晚至28日，开始着手准备并进行实验。

实验环境

- **CPU**：13th Gen Intel(R) Core(TM) i5-13500H 2.6GHz
- **编译器**：g++ 版本 11.3.0
- **操作系统**：Ubuntu 22.04.3 LTS
- **实验设备**：ThinkBook 16 G5+ IRH
- **设备状态**：保持插电状态，清理后台进程

> 由于电脑调度策略等等因素，设备的离电与充电状态可能对性能有极大影响。经过实际测试，实验室用的设备在离电状态下，即使将设备切换为性能模式，CPU依然最多只能跑满70%，且状态极不稳定。充电状态下则可以发挥出全部性能。
>
> 这是编译优化之后我发现的第二个重大优化，曾经困扰我许久。**二者带来的性能差距在2~3倍。**

实验测试

前期的项目设计已经于26日完成并验收，此后就是长期的实验与数据收集。为了大量进行测试、获取更多数据，我为实验设计了一个脚本（`run.py`）来自动化运行。

一共进行了两组测试：

第一组：从整体上，将所有五个版本的实现，针对不同的 N 均进行了 10 轮测试，取平均值得到该条件下该方案的优化效果。数据的结果保存在 `test1_data.xlsx` 下。

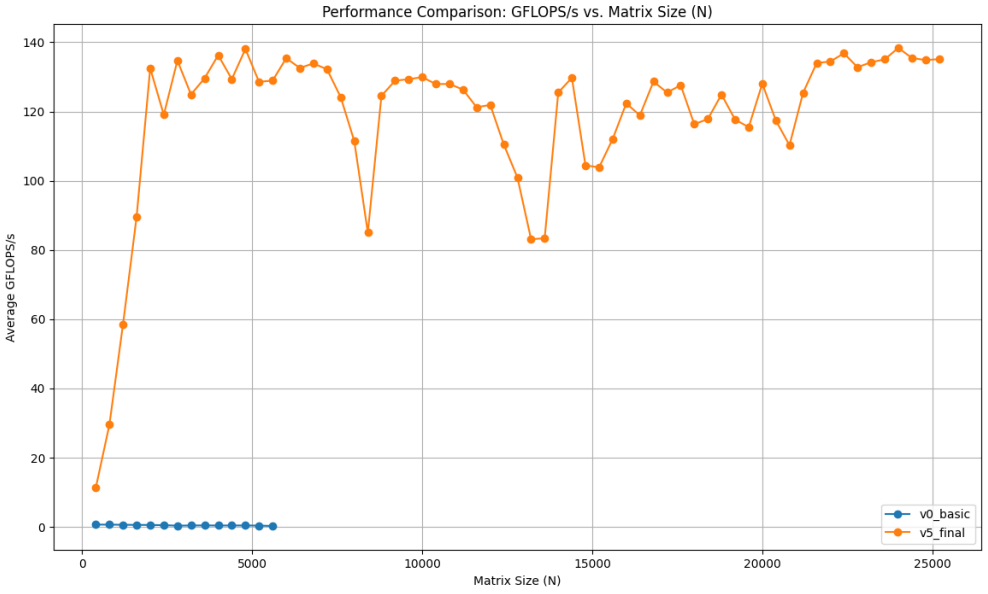
> 为了还原基准状态，在编译 `v0-basic.cc` 时关闭了编译优化

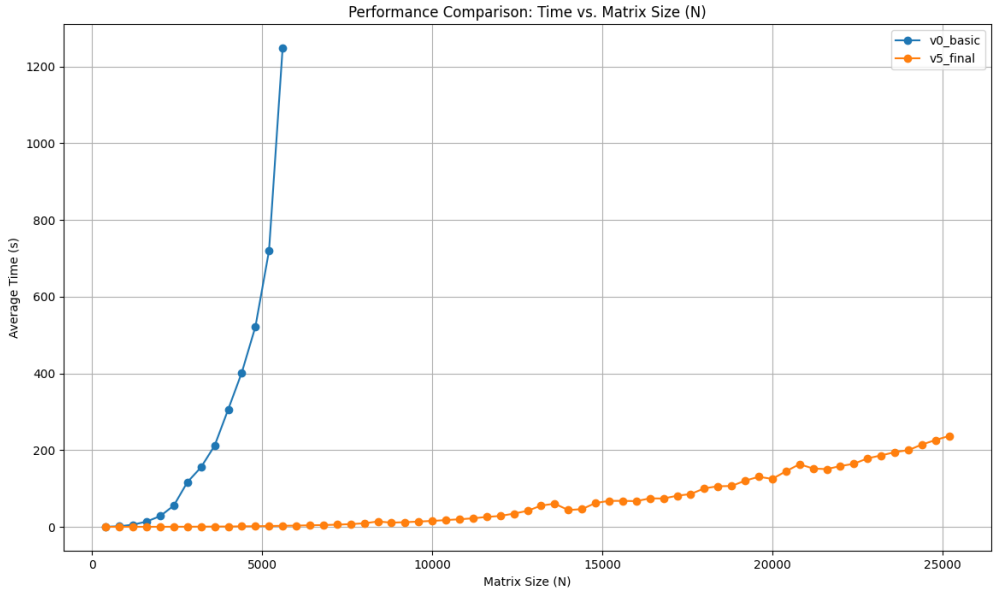
第二组：取最终方案，与基准进行对比测试。将矩阵大小调整为更大范围，最终得到结果。

> 由于时间紧张，我利用了一天半的时间收集数据，期间电脑一直保持高负载工作。本计划将规模扩大到30000*30000，但由于后期数据规模太大、即使是优化后也要大量时间，而最初的版本又非常欠优化、运行时效率极低，导致部分数据残缺。

实验结果与数据分析

实验结果导出了下面两张数据图，分别反映了性能与耗时。

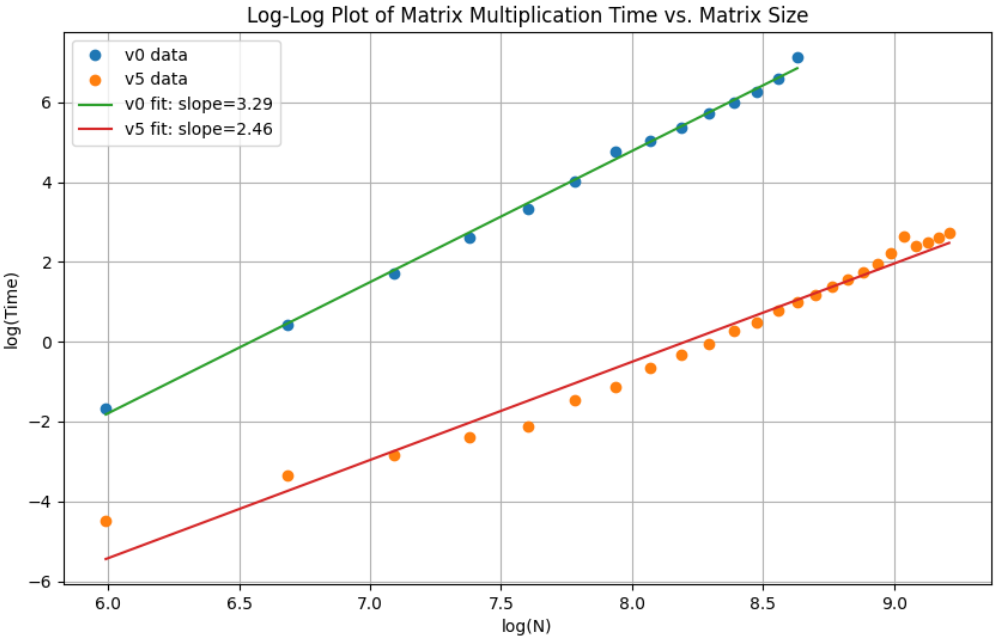




从图1不难看出，**相比于传统解法0.4 GFLOPS/s左右的性能（图上看不出来，但可以从`data.csv`文件中分析），优化方法中，普遍达到100 GFLOPS/s以上、最高140 GFLOPS/s的性能，优势十分显著。**

从理论上讲，矩阵乘法的时间复杂度是立方的，即 O(N^3)。这意味着矩阵的大小 N 增加时，计算时间应该按 N^3 的速度增长。实验结果基本符合这一理论。

而将矩阵分块后，理论上复杂度降为 O(N^2)，又采取了多种优化方法，拟合后的结果显示如图：



实验不足

与理论峰值的差距

与理论峰值性能尚存巨大差距。分析有以下原因：

1. 已经进行的测试存在缺陷。比如第一组数据不及第二组，是因为第一组实验是在离电状态下进行的测试（当时还没有注意到这一问题），结果只能用作大致比较，可参考性几乎为零。
2. 受限于自己对于问题的认识，还有更高阶的优化算法难以实现，性能依然潜力巨大

3. 其他方面的限制，比如内存带宽、工作负载，导致CPU很难发挥出全部性能

干扰引起的数据误差

在获取数据时电脑的其他进程疑似会对实验结果有影响。尽管已经尽可能在实验测试时，通过重新启动、清理进程等方式，降低其他CPU负载，并尽可能避免断电的情况，让CPU更多性能专注于测试，但是依然难免存在干扰。

此外，尽管尽可能保持实验环境稳定，受限于客观条件，一些干扰难以避免。**比如数据中几个比较大的波谷，就是机房、食堂、宿舍之间的路上测得的。**

我怀疑这是导致后期性能出现较大幅度波动的原因，但尚缺乏充分的理论支撑。**如果能够多出1~2天时间，就能够进行更加细致、稳定的重测。这样的数据更具价值，也更能读出更多有效结论。**

项目运行与复现

我编写了一个 Python 脚本 `run.py` 来进行自动化测试。将想要复现测试的文件放在src文件夹中，执行脚本即可。脚本会首先进行编译，编译后的文件存储到bin文件夹下，并运行不同版本的代码，并记录每次运行的结果到 `result/results.csv` 文件中。

> 为了适配脚本并规避可能的性能损耗，矩阵大小作为编译时的参数传入。

脚本中可以根据需要，便捷地进行如下调整：

1. 将想要测试的版本放置到 `src` 目录下即可，会自动识别并编译运行
2. 设置想要测试的矩阵大小 `N`
3. 设置每次相同的测试的循环次数 `loop`，会自动将结果取平均值，并且会记录输出每次测试的所有数据

总结

实验证明，在CPU层面，的确可以通过一系列手段加速计算。**在数据规模较小的时候，优化优势尚没有完全展现；而当数据规模扩大时，优化带来的则是数百倍的性能提升。**我通过一系列优化措施，成功地在矩阵乘法计算中显著提升了计算性能。

这是一次充满波折但又让人欣喜的实验。尽管有 Intel MKL 等等高性能计算库的存在、能够将这一问题进行非常极致的优化，但是尝试从头解决这一问题依然有着非常重大的意义。探索求解的过程，的确让人乐在其中。

参考

作为一个经典问题，网络上有大量关于GEMM计算优化的讨论。但实际上手就会发现，这并不意味着能够照抄一个已有的算法或者解决思路，转瞬间立即获得上百倍的性能提升；同时，AI的智慧也是有限的，只能提供辅助性的工作。**想要真正解决问题，一定是建立在自己对于问题的深入理解与探索上。**

对于未知知识的快速学习，离不开以下几项内容：

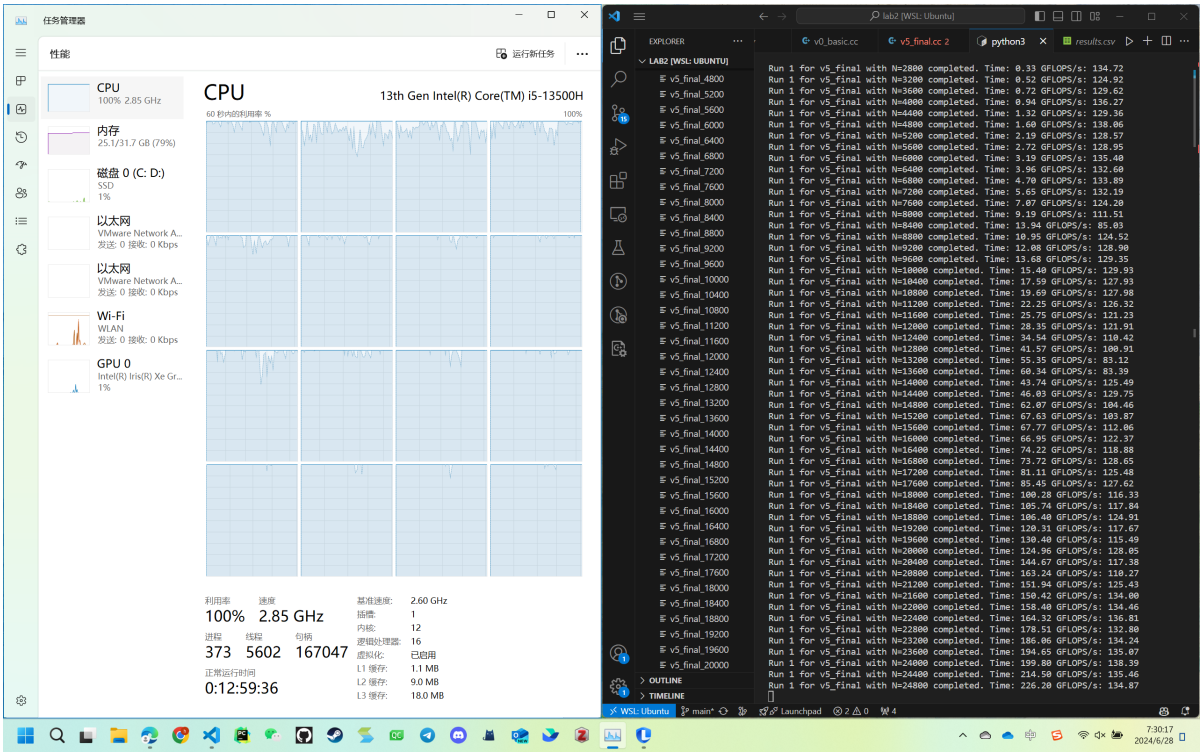
1. 对于CPU架构、包括核心、线程、缓存等等概念的基本认识必不可少，我对此的认识是日积月累的。这些能够让人快速理解“为什么要进行这样反直觉的优化”、“如何将这种思想应用于自己的项目当中”。

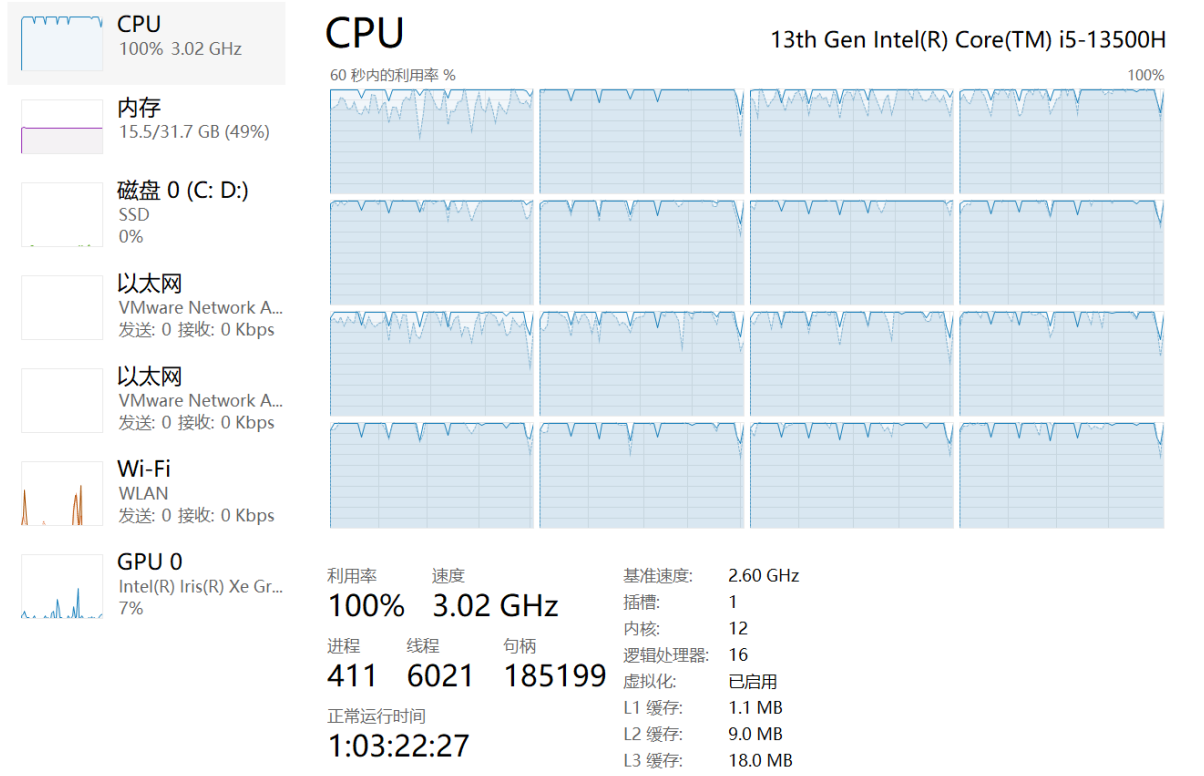
2. 信息检索能力，无论是B站、Google、GitHub，寻找思路要取百家之长。

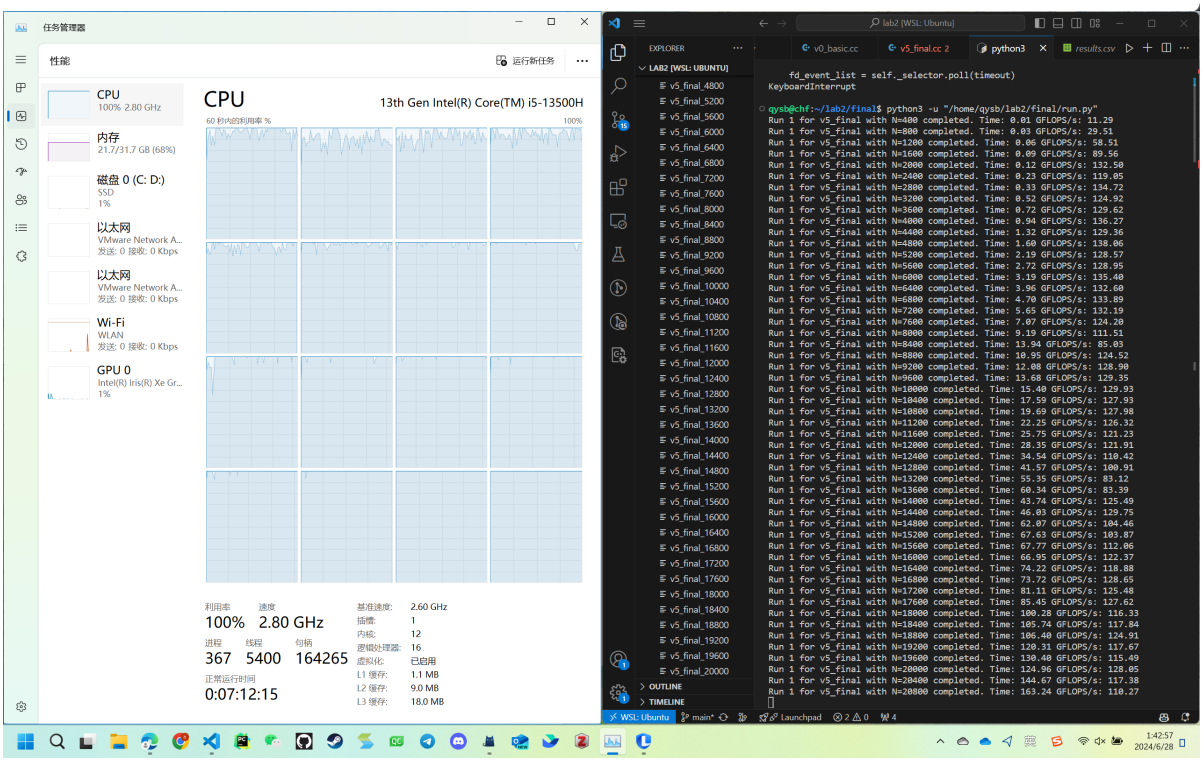
3. `ChatGPT`、`GitHub Copilot`等等AI的支持和辅助，让编程与数据分析效率极大提高。

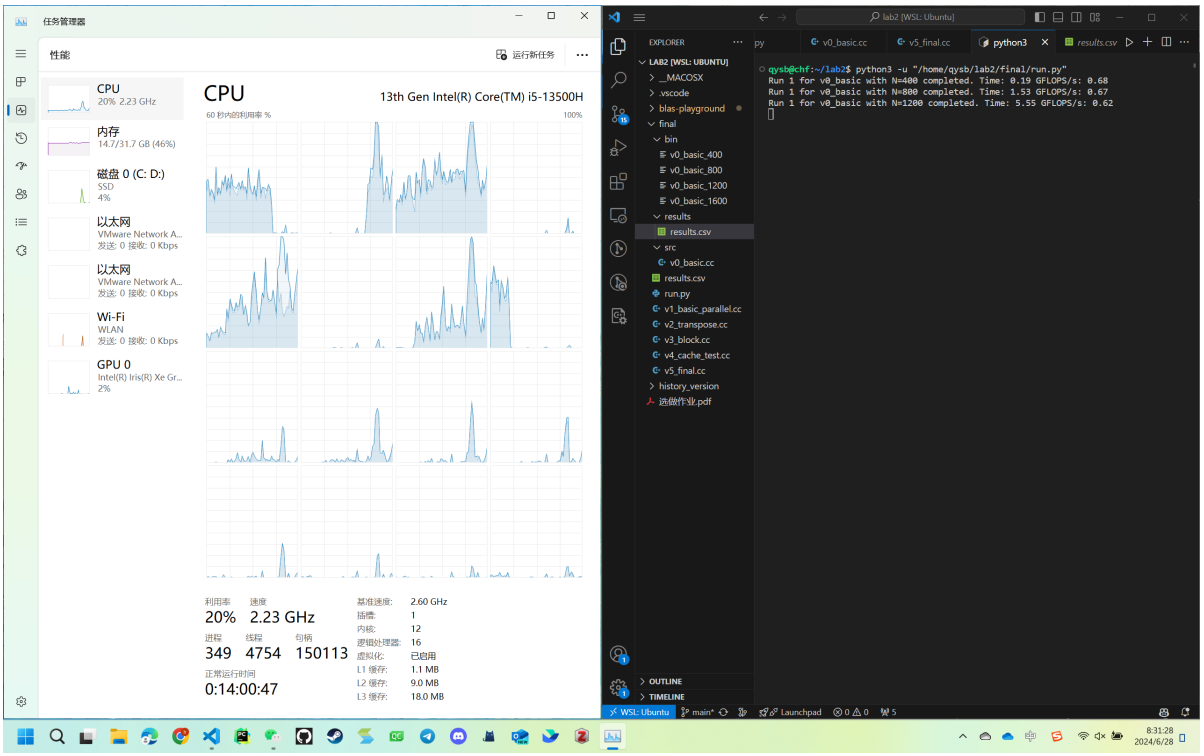
实验截图

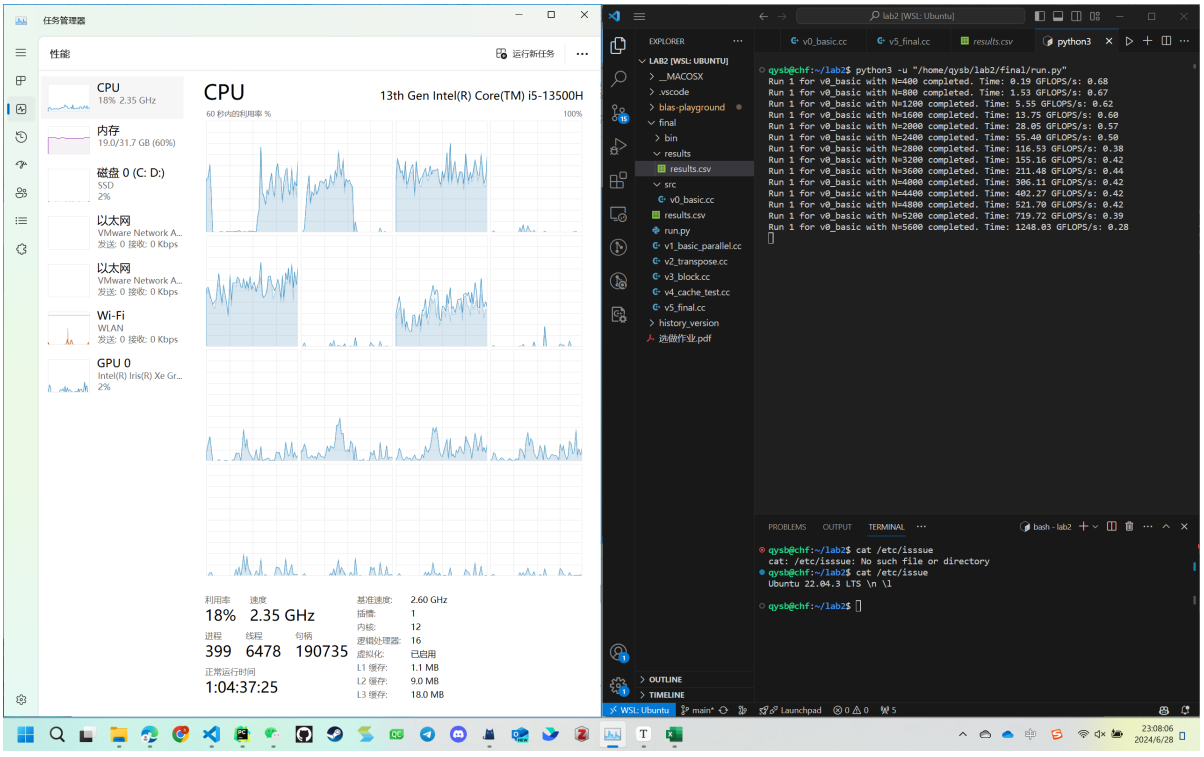
> 未优化情况下，CPU明显无法发挥出全部性能；优化之后，长期保持在95%以上利用率、频率升至3GHz左右。




CPU
100% 3.02 GHz
内存
15.5/31.7 GB (49%)
磁盘 0 (C: D:)
SSD
0%
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
Wi-Fi
WLAN
发送: 0 接收: 0 Kbps
GPU 0
Intel(R) Iris(R) Xe Gr...
7%
CPU
13th Gen Intel(R) Core(TM) i5-13500H
60 秒内的利用率 %
100%
利用率
100%
速度
3.02 GHz
进程
411
线程
6021
句柄
185199
正常运行时间
1:03:22:27
基准速度: 2.60 GHz
插槽: 1
内核: 12
逻辑处理器: 16
虚拟化: 已启用
L1 缓存: 1.1 MB
L2 缓存: 9.0 MB
L3 缓存: 18.0 MB


任务管理器
性能
运行新任务
CPU
13th Gen Intel(R) Core(TM) i5-13500H
利用率
100%
速度
2.80 GHz
进程
367
线程
5400
句柄
164265
正常运行时间
0:07:12:15
基准速度: 2.60 GHz
插槽: 1
内核: 12
逻辑处理器: 16
虚拟化: 已启用
L1 缓存: 1.1 MB
L2 缓存: 9.0 MB
L3 缓存: 18.0 MB
EXPLORER
LAB2 [WSL: UBUNTU]
v0_basic.cc
v5_final.cc 2
python3
results.csv
fd_event_list = self._selector.poll(timeout)
KeyboardInterrupt
qysb@chf:~/lab2/final$ python3 -u "/home/qysb/lab2/final/run.py"
Run 1 for v5_final with N=400 completed. Time: 0.01 GFLOPS/s: 11.29
Run 1 for v5_final with N=800 completed. Time: 0.03 GFLOPS/s: 29.51
Run 1 for v5_final with N=1200 completed. Time: 0.06 GFLOPS/s: 58.51
Run 1 for v5_final with N=1600 completed. Time: 0.09 GFLOPS/s: 89.56
Run 1 for v5_final with N=2000 completed. Time: 0.12 GFLOPS/s: 132.50
Run 1 for v5_final with N=2400 completed. Time: 0.23 GFLOPS/s: 119.05
Run 1 for v5_final with N=2800 completed. Time: 0.33 GFLOPS/s: 134.72
Run 1 for v5_final with N=3200 completed. Time: 0.52 GFLOPS/s: 124.92
Run 1 for v5_final with N=3600 completed. Time: 0.72 GFLOPS/s: 129.62
Run 1 for v5_final with N=4000 completed. Time: 0.94 GFLOPS/s: 136.27
Run 1 for v5_final with N=4400 completed. Time: 1.32 GFLOPS/s: 129.36
Run 1 for v5_final with N=4800 completed. Time: 1.60 GFLOPS/s: 138.06
Run 1 for v5_final with N=5200 completed. Time: 2.19 GFLOPS/s: 128.57
Run 1 for v5_final with N=5600 completed. Time: 2.72 GFLOPS/s: 128.95
Run 1 for v5_final with N=6000 completed. Time: 3.19 GFLOPS/s: 135.40
Run 1 for v5_final with N=6400 completed. Time: 3.96 GFLOPS/s: 132.60
Run 1 for v5_final with N=6800 completed. Time: 4.70 GFLOPS/s: 133.89
Run 1 for v5_final with N=7200 completed. Time: 5.65 GFLOPS/s: 132.19
Run 1 for v5_final with N=7600 completed. Time: 7.07 GFLOPS/s: 124.20
Run 1 for v5_final with N=8000 completed. Time: 9.19 GFLOPS/s: 111.51
Run 1 for v5_final with N=8400 completed. Time: 13.94 GFLOPS/s: 85.03
Run 1 for v5_final with N=8800 completed. Time: 10.95 GFLOPS/s: 124.52
Run 1 for v5_final with N=9200 completed. Time: 12.08 GFLOPS/s: 128.90
Run 1 for v5_final with N=9600 completed. Time: 13.68 GFLOPS/s: 129.35
Run 1 for v5_final with N=10000 completed. Time: 15.40 GFLOPS/s: 129.93
Run 1 for v5_final with N=10400 completed. Time: 17.59 GFLOPS/s: 127.93
Run 1 for v5_final with N=10800 completed. Time: 19.69 GFLOPS/s: 127.98
Run 1 for v5_final with N=11200 completed. Time: 22.25 GFLOPS/s: 126.32
Run 1 for v5_final with N=11600 completed. Time: 25.75 GFLOPS/s: 121.23
Run 1 for v5_final with N=12000 completed. Time: 28.35 GFLOPS/s: 121.91
Run 1 for v5_final with N=12400 completed. Time: 34.54 GFLOPS/s: 110.42
Run 1 for v5_final with N=12800 completed. Time: 41.57 GFLOPS/s: 100.91
Run 1 for v5_final with N=13200 completed. Time: 55.35 GFLOPS/s: 83.12
Run 1 for v5_final with N=13600 completed. Time: 60.34 GFLOPS/s: 83.39
Run 1 for v5_final with N=14000 completed. Time: 43.74 GFLOPS/s: 125.49
Run 1 for v5_final with N=14400 completed. Time: 46.03 GFLOPS/s: 129.75
Run 1 for v5_final with N=14800 completed. Time: 62.07 GFLOPS/s: 104.46
Run 1 for v5_final with N=15200 completed. Time: 67.63 GFLOPS/s: 103.87
Run 1 for v5_final with N=15600 completed. Time: 67.77 GFLOPS/s: 112.06
Run 1 for v5_final with N=16000 completed. Time: 66.95 GFLOPS/s: 122.37
Run 1 for v5_final with N=16400 completed. Time: 74.22 GFLOPS/s: 118.88
Run 1 for v5_final with N=16800 completed. Time: 73.72 GFLOPS/s: 128.65
Run 1 for v5_final with N=17200 completed. Time: 81.11 GFLOPS/s: 125.48
Run 1 for v5_final with N=17600 completed. Time: 85.45 GFLOPS/s: 127.62
Run 1 for v5_final with N=18000 completed. Time: 100.28 GFLOPS/s: 116.33
Run 1 for v5_final with N=18400 completed. Time: 105.74 GFLOPS/s: 117.84
Run 1 for v5_final with N=18800 completed. Time: 106.40 GFLOPS/s: 124.91
Run 1 for v5_final with N=19200 completed. Time: 120.31 GFLOPS/s: 117.67
Run 1 for v5_final with N=19600 completed. Time: 130.40 GFLOPS/s: 115.49
Run 1 for v5_final with N=20000 completed. Time: 124.96 GFLOPS/s: 128.05
Run 1 for v5_final with N=20400 completed. Time: 144.67 GFLOPS/s: 117.38
Run 1 for v5_final with N=20800 completed. Time: 163.24 GFLOPS/s: 110.27
OUTLINE
TIMELINE
WSL: Ubuntu
1:42:57
2024/6/28


任务管理器
性能
运行新任务
CPU
20% 2.23 GHz
内存
14.7/31.7 GB (46%)
磁盘 0 (C: D:)
SSD
4%
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
Wi-Fi
WLAN
发送: 0 接收: 0 Kbps
GPU 0
Intel(R) Iris(R) Xe Gr...
2%
CPU
13th Gen Intel(R) Core(TM) i5-13500H
60 秒内的利用率 %
100%
利用率
20%
速度
2.23 GHz
进程
349
线程
4754
句柄
150113
正常运行时间
0:14:00:47
基准速度: 2.60 GHz
插槽: 1
内核: 12
逻辑处理器: 16
虚拟化: 已启用
L1 缓存: 1.1 MB
L2 缓存: 9.0 MB
L3 缓存: 18.0 MB
EXPLORER
LAB2 [WSL: UBUNTU]
__MACOSX
.vscode
blas-playground
final
bin
v0_basic_400
v0_basic_800
v0_basic_1200
v0_basic_1600
results
results.csv
src
v0_basic.cc
results.csv
run.py
v1_basic_parallel.cc
v2_transpose.cc
v3_block.cc
v4_cache_test.cc
v5_final.cc
history_version
选做作业.pdf
OUTLINE
TIMELINE
v0_basic.cc
v5_final.cc
python3
results.csv
qysb@chf:~/lab2$ python3 -u "/home/qysb/lab2/final/run.py"
Run 1 for v0_basic with N=400 completed. Time: 0.19 GFLOPS/s: 0.68
Run 1 for v0_basic with N=800 completed. Time: 1.53 GFLOPS/s: 0.67
Run 1 for v0_basic with N=1200 completed. Time: 5.55 GFLOPS/s: 0.62
WSL: Ubuntu
main*
Launchpad
8:31:28
2024/6/28


任务管理器
性能
运行新任务
CPU
18% 2.35 GHz
内存
19.0/31.7 GB (60%)
磁盘 0 (C: D:)
SSD
2%
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
以太网
VMware Network A...
发送: 0 接收: 0 Kbps
Wi-Fi
WLAN
发送: 0 接收: 0 Kbps
GPU 0
Intel(R) Iris(R) Xe Gr...
2%
CPU
13th Gen Intel(R) Core(TM) i5-13500H
60 秒内的利用率 %
100%
利用率
18%
速度
2.35 GHz
进程
399
线程
6478
句柄
190735
正常运行时间
1:04:37:25
基准速度: 2.60 GHz
插槽: 1
内核: 12
逻辑处理器: 16
虚拟化: 已启用
L1 缓存: 1.1 MB
L2 缓存: 9.0 MB
L3 缓存: 18.0 MB
EXPLORER
LAB2 [WSL: UBUNTU]
__MACOSX
.vscode
blas-playground
final
bin
results
results.csv
src
v0_basic.cc
results.csv
run.py
v1_basic_parallel.cc
v2_transpose.cc
v3_block.cc
v4_cache_test.cc
v5_final.cc
history_version
选做作业.pdf
OUTLINE
TIMELINE
v0_basic.cc
v5_final.cc
results.csv
python3
qysb@chf:~/lab2$ python3 -u "/home/qysb/lab2/final/run.py"
Run 1 for v0_basic with N=400 completed. Time: 0.19 GFLOPS/s: 0.68
Run 1 for v0_basic with N=800 completed. Time: 1.53 GFLOPS/s: 0.67
Run 1 for v0_basic with N=1200 completed. Time: 5.55 GFLOPS/s: 0.62
Run 1 for v0_basic with N=1600 completed. Time: 13.75 GFLOPS/s: 0.60
Run 1 for v0_basic with N=2000 completed. Time: 28.05 GFLOPS/s: 0.57
Run 1 for v0_basic with N=2400 completed. Time: 55.40 GFLOPS/s: 0.50
Run 1 for v0_basic with N=2800 completed. Time: 116.53 GFLOPS/s: 0.38
Run 1 for v0_basic with N=3200 completed. Time: 155.16 GFLOPS/s: 0.42
Run 1 for v0_basic with N=3600 completed. Time: 211.48 GFLOPS/s: 0.44
Run 1 for v0_basic with N=4000 completed. Time: 306.11 GFLOPS/s: 0.42
Run 1 for v0_basic with N=4400 completed. Time: 402.27 GFLOPS/s: 0.42
Run 1 for v0_basic with N=4800 completed. Time: 521.70 GFLOPS/s: 0.42
Run 1 for v0_basic with N=5200 completed. Time: 719.72 GFLOPS/s: 0.39
Run 1 for v0_basic with N=5600 completed. Time: 1248.03 GFLOPS/s: 0.28
PROBLEMS
OUTPUT
TERMINAL
bash - lab2
qysb@chf:~/lab2$ cat /etc/isssue
cat: /etc/isssue: No such file or directory
qysb@chf:~/lab2$ cat /etc/issue
Ubuntu 22.04.3 LTS \n \l
qysb@chf:~/lab2$
WSL: Ubuntu
main*
Launchpad
23:08:06
2024/6/28