实验报告

课程：高性能计算应用实践

姓名：王峻阳

学号：220110317

学院：计算机科学与技术学院

学期：2023 年秋季学期

实验日期：2023 年 10 月 7 日

一、实验环境

操作系统　Ubuntu 22.04.2 LTS（在 WSL2 上运行）

内存　7808.3 MiB（分配给 WSL2 的部分）

CPU　名称：12th Gen Intel(R) Core(TM) i5-12500H
指令集架构：x86-64
字长：64 位
物理核数：12 核，其中 4 核是性能核，8 核是能效核
线程数：16 线程，其中 8 线程来自性能核，8 线程来自能效核
基准频率：2.50 GHz（性能核）、1.8 GHz（能效核）
最大睿频频率：4.50 GHz（性能核）、3.30 GHz（能效核）

CPU 缓存　性能核：
L1d　4 * 48 KiB (12-way, 64-byte line)
L1i　4 * 32 KiB (8-way, 64-byte line)
L2　4 * 1.25 MiB (10-way, 64-byte line)
能效核：
L1d　8 * 32 KiB (8-way, 64-byte line)
L1i　8 * 64 KiB (8-way, 64-byte line)
L2　2 * 2 MiB (16-way, 64-byte line)

L3　18 MiB (12-way, 64-byte line)

二、Gflops 理论峰值计算

一、公式

$$\mathrm{FLOPS}_{\mathrm{core}} = \frac{\mathrm{instructions}}{\mathrm{cycle}} \times \frac{\mathrm{operations}}{\mathrm{instruction}} \times \frac{\mathrm{FLOPs}}{\mathrm{operation}} \times \frac{\mathrm{cycles}}{\mathrm{second}}$$

二、数值

1. cycles/second: Intel(R) Core(TM) i5-12500H 有 12 个核，其性能核和能效核的频率不同，要分开计算。由于其频率会受控变化，下面分别以基准频率和最高睿频频率计算。

	基准频率	最大睿频频率
性能核	2.50 GHz	4.50 GHz
能效核	1.80 GHz	3.30 GHz

2. instructions/cycle: 每个周期能执行的指令数由执行单元数决定，这里将 AVX2 & FMA 的执行单元数视作 2 个。

3. operations/instruction: 使用 AVX2 指令集，一条指令能操作 4 组双精度浮点数。

4. FLOPS/operation: 使用 FMA 指令集，一个操作能完成双精度浮点数的乘法计算和加法计算，也就是 2 次浮点数计算。

三、计算

按基准频率：

$$\mathrm{Gflops} = 4 \times 2.50 \times 16 + 8 \times 1.80 \times 16 = 390.4$$

按最大睿频频率：

$$\mathrm{Gflops} = 4 \times 4.50 \times 16 + 8 \times 3.30 \times 16 = 710.4$$

二、软件依赖

一、编译器：
gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0

二、MPI
MPICH (Version 3.3.2)

三、BLAS
OpenBLAS (0.3.24.dev)

三、参数调优

一、问题规模

根据 https://www.netlib.org/benchmark/hpl/faqs.html 的说明，问题规模最好在空闲内存范围之内，并且越大越好，得到规模在 25000 附近为佳。

二、分块大小

虽然根据上述 FAQ 页面，分块大小一般都在 32~256 之间，但发现在很大范围内分块大小的对 Gflops 的影响并不显著。

三、P x Q

由于只有一个节点，设置 P=1, Q=1。

四、测试结果

```
N      :   25000
NB     :     500
PMAP   : Row-major process mapping
P      :       1
Q      :       1
PFACT  :   Right
NBMIN  :       4
NDIV   :       2
RFACT  :   Crout
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be              1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4       25000   500     1     1             55.32             1.8831e+02
```

实际上，由于运算时间太长时 CPU 发热会降低频率，发现在问题规模较小时 Gflops 表现更好：

```
N      :   15000
NB     :     400
PMAP   : Row-major process mapping
P      :       1
Q      :       1
PFACT  :   Right
NBMIN  :       4
NDIV   :       2
RFACT  :   Crout
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be              1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4       15000   400     1     1              10.70             2.1035e+02
```

三、问题和解决过程

一、找不到 MPICH 的安装位置

在网上搜索，发现 apt get 安装的包一般可以在/usr/include 和/usr/lib/下找到，最后发现头文件是在/usr/include/x86_64-linux-gnu/mpich/下，库文件是在/usr/lib/x86_64-linux-gnu/下。另外学习到可以用 dpkg -S 指令找到库文件在哪儿。

二、报错提示 undefined reference to `MAIN_'和 multiple definition of `main'。

原因是 Make.<arch>文件中将 LINKER 设置为了 mpif77，而 mpif77 是 Fortran 的链接器。