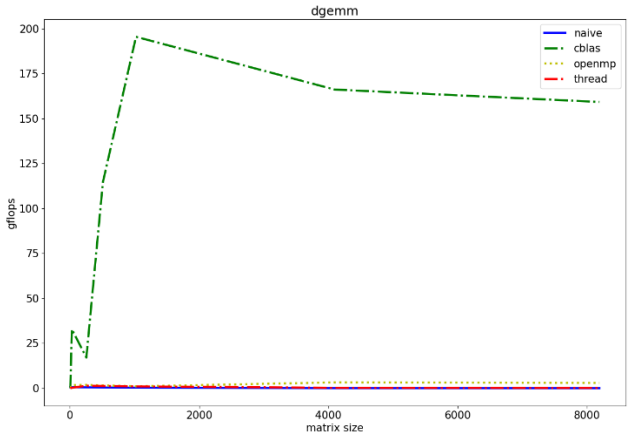
高性能计算应用实践实验报告（lab6）

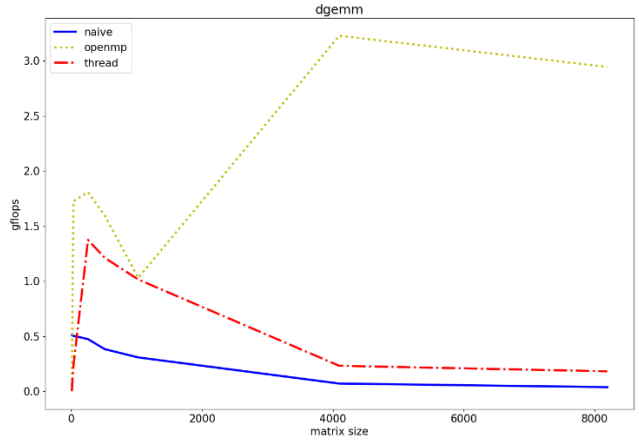
2024 秋 苏涵 2023311B25

实验环境

OS：Linux suh 5.15.153.1-microsoft-standard-WSL2 #1 SMP Fri Mar 29 23:14:13 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
Ubuntu 22.04.3 LTS
gcc：(Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
CPU：型号 11th Gen Intel(R) Core(TM) i7-11370H 频率 3.30GHz 核数 4

数据分析





当矩阵规模为 8*8 时，naive 的 gflops 大于其它方法，其它情况下 gflops 从大到小排列依次为 cblas、openmp、thread、naive。且 cblas 的 gflops 远大于其他三种方法。

实现方式介绍与核心代码

naive

使用三重嵌套循环来计算矩阵乘法。外层两个循环遍历结果矩阵 C 的每个元素，内层循环计算对应元素的和

naive_gemm.c > ...

```c
#include "defs.h"

void MY_MMult(int m, int n, int k, double *a, int lda,
              double *b, int ldb,
              double *c, int ldc)
{
  int i, j, p;

  for (i = 0; i < m; i++) /* Loop over the rows of C */
  {
    for (j = 0; j < n; j++) /* Loop over the columns of C */
    {
      for (p = 0; p < k; p++)
      {
        C(i, j) = C(i, j) + A(i, p) * B(p, j);
      }
    }
  }
}
```

thread

将矩阵乘法任务分配到多个线程上，每个线程负责计算矩阵的一部分，最终合并结果

pthreads_gemm.c > ...

```c
#include "defs.h"
#include <pthread.h>

typedef struct {
    int start_row;
    int end_row;
    double* A;
    double* B;
    double* C;
    int m;
    int n;
} ThreadData;


void* matrix_multiply(void* arg) {
    ThreadData* data = (ThreadData*)arg;
    int start_row = data->start_row;
    int end_row = data->end_row;
    double* A = data->A;
    double* B = data->B;
    double* C = data->C;
    int n = data->n;

    for (int i = start_row; i < end_row; i++) {
        for (int j = 0; j < n; j++) {
            for (int k = 0; k < n; k++) {
                C[i*n+j] = C[i*n+j] + A[i*n+k] * B[k*n+j];
            }
        }
    }
    pthread_exit(NULL);
}

void MY_MMult(int m, int n, int k, double *a, int lda,
              double *b, int ldb,
              double *c, int ldc)
{
    int rows_per_thread = m / NUM_THREADS;
    pthread_t threads[NUM_THREADS];
    ThreadData thread_data[NUM_THREADS];

     for (int i = 0; i < NUM_THREADS;i++) {
            thread_data[i].start_row = i * rows_per_thread;
            thread_data[i].end_row = (i + 1) * rows_per_thread;
            thread_data[i].A = a;
            thread_data[i].B = b;
            thread_data[i].C = c;
            thread_data[i].n = n;
            pthread_create(&threads[i], NULL, matrix_multiply, (void*)&thread_data[i]);
        }

        for (int i = 0; i < NUM_THREADS; i++) {
            pthread_join(threads[i], NULL);
        }
}
```

cblas

使用 BLAS 库中的 cblas_dgemm 函数，自动处理矩阵乘法的计算

```c
// cblas_gemm.c > ...
#include <cblas.h>
#include <stdio.h>

void MY_MMult(int m, int n, int k, double *a, int lda,
              double *b, int ldb,
              double *c, int ldc)
{
  double alpha = 1.0;
  double beta = 1.0;

  cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, m, n, k, alpha, a, lda, b, ldb, beta, c, ldc);

}
```

openmp

使用 OpenMP 并行化矩阵乘法，通过添加指令，利用多核处理器加速计算

```c
// openmp_gemm.c > ...
#include "defs.h"
#include<omp.h>

void MY_MMult(int m, int n, int k, double *a, int lda,
              double *b, int ldb,
              double *c, int ldc)
{
    #pragma omp parallel for
    for (int i = 0; i < m; i++) /* Loop over the rows of C */
    {
        for (int j = 0; j < n; j++) /* Loop over the columns of C */
        {
            for (int p = 0; p < k; p++)
            {
                C(i, j) = C(i, j) + A(i, p) * B(p, j);
            }
        }
    }
}
```

lab3 问题

(1)

```
echo "version = 'naive_gemm';"
```

可通过查看 makefile 中 new 或运行时输出 version 判断调用的是哪个函数

```
$(BUILD_DIR)/test_MMult.x: $(OBJS) defs.h
    $(LINKER) $(OBJS) $(LDFLAGS) -o $@
```

```
OBJS  := $(BUILD_DIR)/util.o $(BUILD_DIR)/REF_MMult.o $(BUILD_DIR)/test_MMult.o $(BUILD_DIR)/$(NEW).o
```

```
NEW  := naive_gemm
```

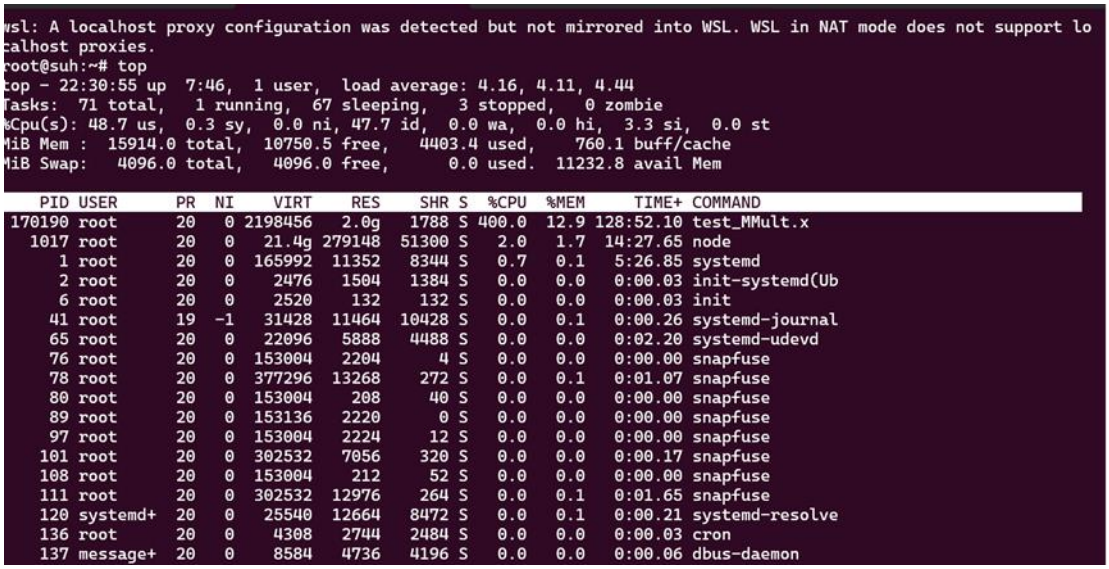
图中代码指定了使用 new 中的 MY_MMult 函数

(2)

```
    @echo "date = '`date`';" > $(DATA_DIR)/output_$(NEW).m
    @echo "version = '$(NEW)';" >> $(DATA_DIR)/output_$(NEW).m
    $(BUILD_DIR)/test_MMult.x >> $(DATA_DIR)/output_$(NEW).m
```

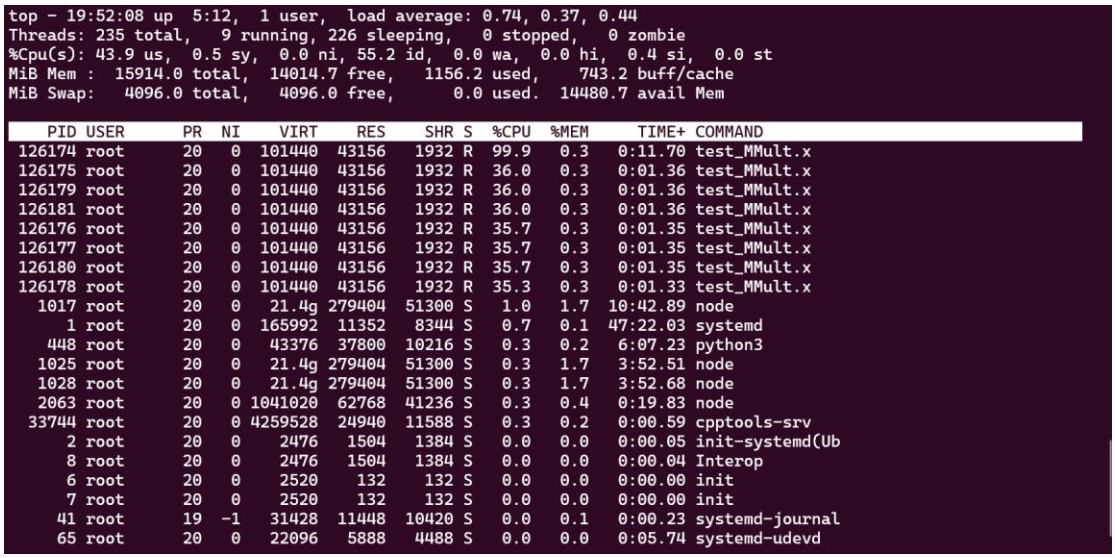
通过 makefile 中图中代码，将运行后的数据写入到_data/output_MMult0.m

thread 运行截图


wsl: A localhost proxy configuration was detected but not mirrored into WSL. WSL in NAT mode does not support lo
calhost proxies.
root@suh:~# top
top - 22:30:55 up 7:46, 1 user, load average: 4.16, 4.11, 4.44
Tasks: 71 total,   1 running, 67 sleeping,   3 stopped,   0 zombie
%Cpu(s): 48.7 us, 0.3 sy, 0.0 ni, 47.7 id, 0.0 wa, 0.0 hi, 3.3 si, 0.0 st
MiB Mem : 15914.0 total, 10750.5 free,   4403.4 used,   760.1 buff/cache
MiB Swap: 4096.0 total,   4096.0 free,   0.0 used. 11232.8 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
170190 root 20 0 2198456 2.0g 1788 S 400.0 12.9 128:52.10 test_MMult.x
1017 root 20 0 21.4g 279148 51300 S 2.0 1.7 14:27.65 node
1 root 20 0 165992 11352 8344 S 0.7 0.1 5:26.85 systemd
2 root 20 0 2476 1504 1384 S 0.0 0.0 0:00.03 init-systemd(Ub
6 root 20 0 2520 132 132 S 0.0 0.0 0:00.03 init
41 root 19 -1 31428 11464 10428 S 0.0 0.1 0:00.26 systemd-journal
65 root 20 0 22096 5888 4488 S 0.0 0.0 0:02.20 systemd-udevd
76 root 20 0 153004 2204 4 S 0.0 0.0 0:00.00 snapfuse
78 root 20 0 377296 13268 272 S 0.0 0.1 0:01.07 snapfuse
80 root 20 0 153004 208 40 S 0.0 0.0 0:00.00 snapfuse
89 root 20 0 153136 2220 0 S 0.0 0.0 0:00.00 snapfuse
97 root 20 0 153004 2224 12 S 0.0 0.0 0:00.00 snapfuse
101 root 20 0 302532 7056 320 S 0.0 0.0 0:00.17 snapfuse
108 root 20 0 153004 212 52 S 0.0 0.0 0:00.00 snapfuse
111 root 20 0 302532 12976 264 S 0.0 0.1 0:01.65 snapfuse
120 systemd+ 20 0 25540 12664 8472 S 0.0 0.1 0:00.21 systemd-resolve
136 root 20 0 4308 2744 2484 S 0.0 0.0 0:00.03 cron
137 message+ 20 0 8584 4736 4196 S 0.0 0.0 0:00.06 dbus-daemon


root@suh:~# pstree
systemd─┬─2*[agetty]
├─cron
├─dbus-daemon
├─init-systemd(Ub─┬─SessionLeader─┬─Relay(940)─┬─4*[cpptools-srv─┬─7*[{cpptools-srv}]]
└─sh─┬─sh─┬─sh─┬─node─┬─node─┬─12*[{node}]
├─node─┬─cpptools─┬─17*[{cpptools}+
├─node─┬─6*[{node}]
├─pet
├─python3─┬─python3
└─2*[{python3}]
└─11*[{node}]
├─node─┬─bash
└─11*[{node}]
└─10*[{node}]
├─SessionLeader─┬─Relay(127843)─┬─bash─┬─test_MMult.x
├─2*[test_MMult.x─┬─7*[{test_MMult.x}]]
└─test_MMult.x─┬─4*[{test_MMult.x}]
├─SessionLeader─┬─Relay(128164)─┬─bash─┬─top
├─SessionLeader─┬─Relay(128229)─┬─bash─┬─pstree
├─SessionLeader─┬─Relay(161084)─┬─node─┬─6*[{node}]
├─SessionLeader─┬─Relay(161095)─┬─node─┬─6*[{node}]
├─init─┬─{init}
├─login─┬─bash
└─{init-systemd(Ub}
├─networkd-dispat
├─packagekitd─┬─2*[{packagekitd}]
├─polkitd─┬─2*[{polkitd}]
├─rsyslogd─┬─3*[{rsyslogd}]
├─snapd─┬─13*[{snapd}]
├─5*[snapfuse─┬─2*[{snapfuse}]]
├─snapfuse─┬─5*[{snapfuse}]
├─2*[snapfuse─┬─4*[{snapfuse}]]
├─subiquity-serve─┬─python3.10─┬─python3
└─5*[{python3.10}]
├─systemd─┬─(sd-pam)
├─systemd-journal
├─systemd-logind
├─systemd-resolve
├─systemd-udevd
└─unattended-upgr─┬─{unattended-upgr}

openmp 运行截图



```
top - 19:52:08 up  5:12,  1 user,  load average: 0.74, 0.37, 0.44
Threads: 235 total,   9 running, 226 sleeping,   0 stopped,   0 zombie
%Cpu(s): 43.9 us,  0.5 sy,  0.0 ni, 55.2 id,  0.0 wa,  0.0 hi,  0.4 si,  0.0 st
MiB Mem :  15914.0 total,  14014.7 free,   1156.2 used,    743.2 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  14480.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 126174 root      20   0  101440  43156   1932 R  99.9   0.3   0:11.70 test_MMult.x
 126175 root      20   0  101440  43156   1932 R  36.0   0.3   0:01.36 test_MMult.x
 126179 root      20   0  101440  43156   1932 R  36.0   0.3   0:01.36 test_MMult.x
 126181 root      20   0  101440  43156   1932 R  36.0   0.3   0:01.36 test_MMult.x
 126176 root      20   0  101440  43156   1932 R  35.7   0.3   0:01.35 test_MMult.x
 126177 root      20   0  101440  43156   1932 R  35.7   0.3   0:01.35 test_MMult.x
 126180 root      20   0  101440  43156   1932 R  35.7   0.3   0:01.35 test_MMult.x
 126178 root      20   0  101440  43156   1932 R  35.3   0.3   0:01.33 test_MMult.x
   1017 root      20   0   21.4g 279404  51300 S   1.0   1.7  10:42.89 node
      1 root      20   0  165992  11352   8344 S   0.7   0.1  47:22.03 systemd
    448 root      20   0   43376  37800  10216 S   0.3   0.2   6:07.23 python3
   1025 root      20   0   21.4g 279404  51300 S   0.3   1.7   3:52.51 node
   1028 root      20   0   21.4g 279404  51300 S   0.3   1.7   3:52.68 node
   2063 root      20   0 1041020  62768  41236 S   0.3   0.4   0:19.83 node
  33744 root      20   0 4259528  24940  11588 S   0.3   0.2   0:00.59 cpptools-srv
      2 root      20   0    2476   1504   1384 S   0.0   0.0   0:00.05 init-systemd(Ub
      8 root      20   0    2476   1504   1384 S   0.0   0.0   0:00.04 Interop
      6 root      20   0    2520    132    132 S   0.0   0.0   0:00.00 init
      7 root      20   0    2520    132    132 S   0.0   0.0   0:00.00 init
     41 root      19  -1   31428  11448  10420 S   0.0   0.1   0:00.23 systemd-journal
     65 root      20   0   22096   5888   4488 S   0.0   0.0   0:05.74 systemd-udevd
```

```
systemd─┬─2*[agetty]
        ├─cron
        ├─dbus-daemon
        ├─init-systemd(Ub─┬─SessionLeader───Relay(940)─┬─4*[cpptools-srv───7*[{cpptools-srv}]]
        │                 │                           └─sh───sh───sh───node─┬─node───12*[{node}]
        │                 │                                                 ├─node─┬─cpptools───17*[{cpptools}+
        │                 │                                                 │      ├─node───6*[{node}]
        │                 │                                                 │      ├─pet
        │                 │                                                 │      ├─python3─┬─python3
        │                 │                                                 │      │         └─2*[{python3}]
        │                 │                                                 │      └─11*[{node}]
        │                 │                                                 ├─node─┬─bash
        │                 │                                                 │      └─11*[{node}]
        │                 │                                                 └─10*[{node}]
        │                 ├─SessionLeader───Relay(91608)───node───6*[{node}]
        │                 ├─SessionLeader───Relay(91618)───node───6*[{node}]
        │                 ├─SessionLeader───Relay(112933)───bash───test_MMult.x───7*[{test_MMult.x}]
        │                 ├─SessionLeader───Relay(114129)───bash───pstree
        │                 ├─SessionLeader───Relay(116605)───bash───top
        │                 ├─init───{init}
        │                 ├─login───bash
        │                 └─{init-systemd(Ub}
        ├─networkd-dispat
        ├─packagekitd───2*[{packagekitd}]
        ├─polkitd───2*[{polkitd}]
        ├─rsyslogd───3*[{rsyslogd}]
        ├─snapd───13*[{snapd}]
        ├─5*[snapfuse───2*[{snapfuse}]]
        ├─snapfuse───5*[{snapfuse}]
        ├─2*[snapfuse───4*[{snapfuse}]]
        ├─subiquity-serve───python3.10─┬─python3
        │                              └─5*[{python3.10}]
        ├─systemd───(sd-pam)
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-resolve
        ├─systemd-udevd
        └─unattended-upgr───{unattended-upgr}
```