

高性能计算应用实践

Lab5 实验报告

朱祉睿 220110501 大二秋

1. 实验方案

将矩阵 C 分成两块，上半块和下半块，创建两个线程分别进行运算。

2. 实验环境

OS: Vmware 虚拟机下 Linux Ubuntu 22.04.2 LTS

gcc 版本: 11.4.0

CPU : AMD® Ryzen 7 5800h with radeon graphics

核数: 8/16 (虚拟机中只有两核)

内存: 16 GB (虚拟机为 4GB)

3. 关键代码

```

#include <assert.h>
#include <pthread.h>
#include <unistd.h>
typedef struct
{
    double *A;
    double *B;
    double *C;
    int alow;
    int ahigh;
    int blow;
    int bhigh;
    int k;
    int n;
}agv;

void *pthread(void *arg)
{
    agv *P = (agv *)arg;
    for (int i = P->alow-1; i < P->ahigh; i++)
    {
        for (int j = P->blow; j < P->bhigh; j++)
        {
            P->C[i*P->k+j] = 0;
            for (int p = 0; p < P->k; p++)
            {
                P->C[i*P->k+j] += P->A[i*P->k+p] * P->B[p*P->n+j];
            }
        }
    }
}

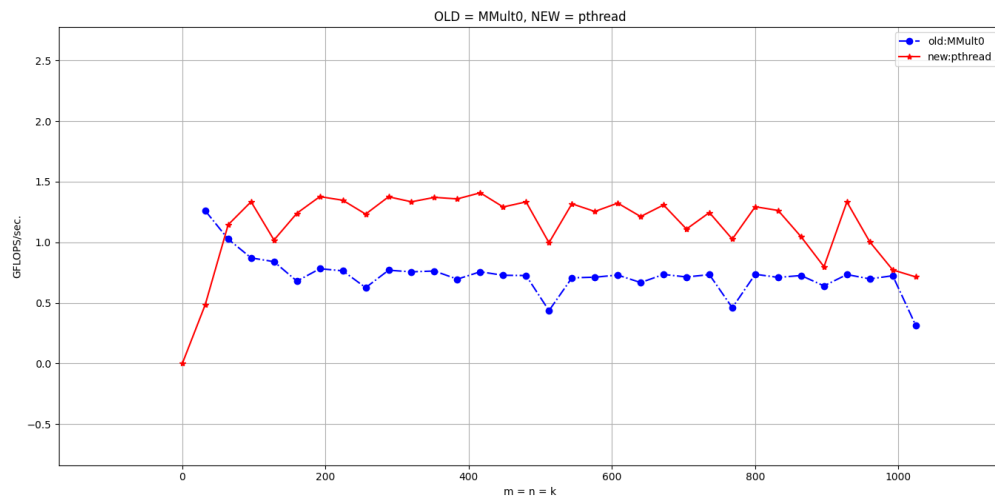
```

```

void MY_MMult( int m, int n, int k, double *a, int lda,
               double *b, int ldb,
               double *c, int ldc )
{
    int i=m/2;
    int rc;
    pthread_t  p1, p2;
    agv c1={a,b,c,1,i,1,n,k,n};
    agv c2={a,b,c,i+1,m,1,n,k,n};
    rc = pthread_create(&p1, NULL, pthread, &c1); assert(rc == 0);
    rc = pthread_create(&p2, NULL, pthread, &c2); assert(rc == 0);
    rc = pthread_join(p1, NULL); assert(rc == 0);
    rc = pthread_join(p2, NULL); assert(rc == 0);
}

```

4.运行结果



5.数据分析

运用 pthread 库多线程进行矩阵乘，由于虚拟机 cpu 核数为 2，所以只创建了两个线程，计算速度也约等于朴素矩阵乘的两倍。

6.碰到的问题

在运用 pthread_create 传参时，未把传递函数的参数改为无类型指针，导致错误，修改后解决。