天津大学
Tianjin University

《并行计算》实验指导书

2024-2025 学年第二学期

指导老师：于策 毕重科 汤善江

实验指导：肖健 杨斌 孙超

《并行计算》实验指导书

2024-2025 学年第二学期

一、实验要求及评分标准

本课程实验目的为提升学生对并行计算的理解认识，培养学生编写基本并行程序的能力，加深对多线程(Phtread)和多进程(MPI)等并行编程的理解认识。

实验课程需要上交实验报告，报告评分标准如下：

实验	内容要求	评分比例	占总分比例
实验一	实验内容	10%	10%
	实验原理	10%	
	程序流程图	30%	
	实验结果及分析	40%	
	实验总结	10%	
实验二	实验内容	10%	30%
	实验原理	10%	
	程序流程图	30%	
	实验结果及分析	40%	
	实验总结	10%	
实验三	实验内容	10%	30%
	实验原理	10%	
	程序流程图	30%	
	实验结果及分析	40%	
	实验总结	10%	
实验四	实验内容	10%	30%
	实验原理	20%	
	设计实现	30%	
	实验结果	30%	
	实验总结	10%	

其中，实验原理包括：**实验数学计算模型**和**实现方法**；编程语言要求使用 C/C++，并行程序设计部分需要给出**程序设计思路**和**伪代码**或**关键部分流程图**；实验结果及分析应包括：**实验结果数据列表**、**加速比曲线**和**实验结果分析**。

二、实验环境介绍及使用方法

1. 集群登录及所需软件

国家超级计算天津中心提供了集成客户端——青索客户端，使用手册见：https://www.nscc-tj.cn/file/青索安装与入门手册.pdf，以下为常规使用推荐的软件：

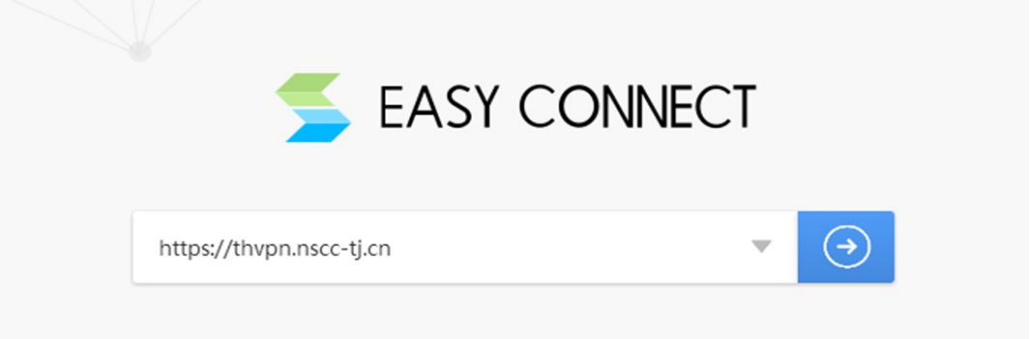
- Easy Connect（VPN 登陆）
  - 校内下载地址：https://vpn.tju.edu.cn/com/EasyConnectInstaller.exe
- SSH（命令行远程登陆命令）
  - Win10 可直接使用 ssh 命令
  - Win7 需要安装 OpenSSH，官方网站：https://www.mls-software.com/opensshd.html
  - 支持 ssh 的独立应用程序
- FileZilla（远程文件传输工具）
  - 官方网站：https://filezilla-project.org/index.php

实验环境不支持图形界面。

PS：Linux 可以安装 Easy Connect（https://vpn.tju.edu.cn 可下载）后直接使用 OpenSSH 进行登录，Mac OS 也可以使用对应的 ssh 命令登录到集群。

2. 登陆 VPN

- 登录 VPN 方式如图所示



在输入框中输入：https://thvpn.nscc-tj.cn，点击右侧箭头连接国家超级计算天津中心 VPN。

在输入框中输入 VPN 登陆的用户名和密码（另行通知），点击登陆按钮等待连接成功。

※注意：输入 VPN 密码时**务必仔细**，可复制粘贴，错误之后会锁定账号，无法登陆。

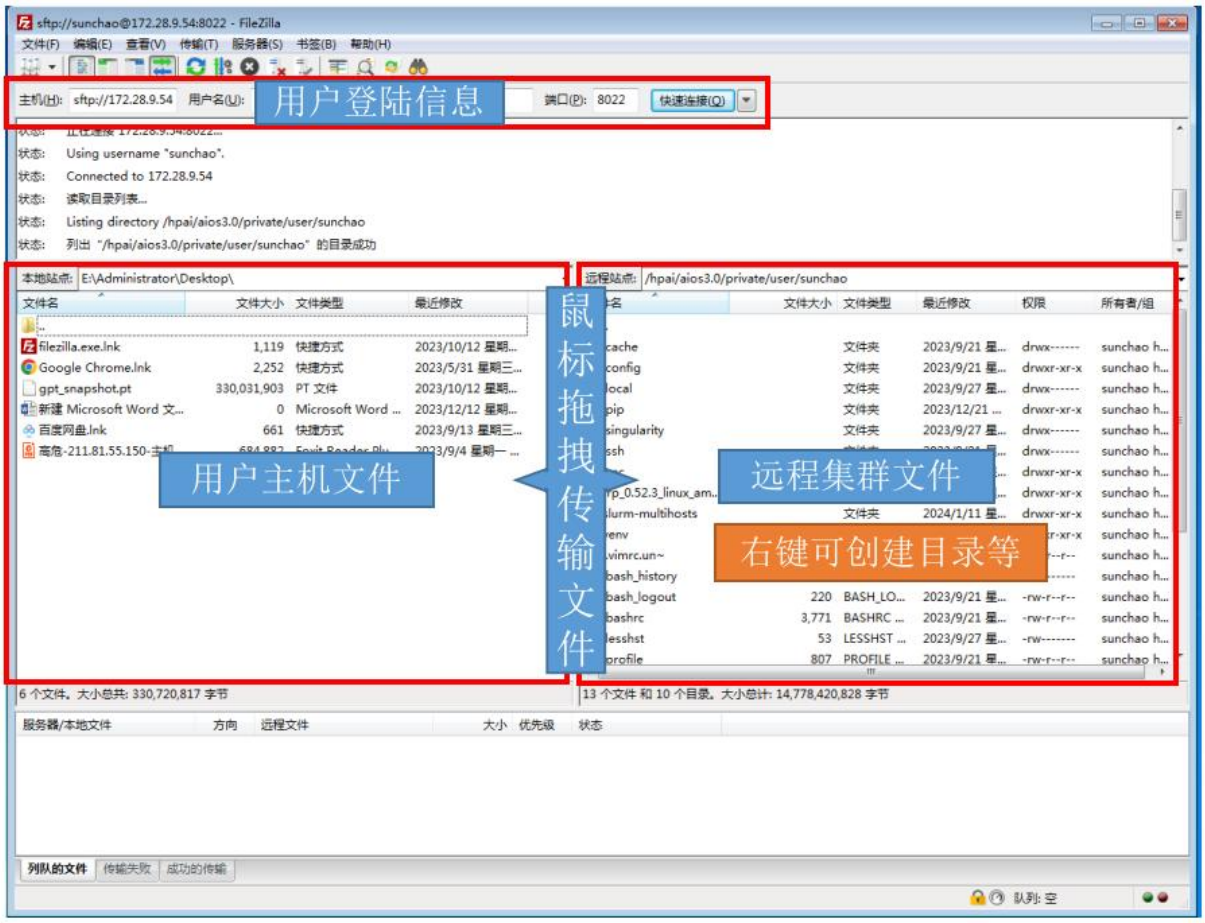
3. 登录集群

- 登录集群方式如图所示：

在提示符后输入：ssh USERNAME@192.168.10.10，其中，需要将 USERNAME 替换为自己的账户名称（另行通知），然后按回车键登录系统。首次登陆会提示密钥指纹信息，输入 yes 继续连接，之后根据提示输入密码（无回显），按回车键确认，待提示欢迎信息后就可以正式使用了。

- 文件传输：

打开 FileZilla 软件，输入主机(sftp://192.168.10.10)、用户名、密码，点击快速连接按扭可以打开文件传输界面，通过鼠标拖动操作即可实现文件传输。



3. 常用 Linux 命令

- ls：列出当前文件夹下文件。如：ls –al
- mkdir：新建文件夹。如：mkdir data
- cd：切换工作文件夹。如：cd data/
- pwd：查看当前文件夹绝对路径
- rm：删除文件或文件夹（需要加上 –r 参数）
- time：获取程序运行时间
- passwd：修改登录密码
- exit：退出登录

4. 实验环境

i. 系统参数

- 国家超级计算天津中心定制操作系统
- 使用国产飞腾处理器
- 天河自主高速互联网络（400GB/s）

- 单核理论性能（双精度）9.2 GFlops
- 单节点理论性能（双精度）588.8 GFlops

ii. 编译环境

- GCC 9.3.0：`gcc`，`g++`，`gfort` 等
- OpenMPI 4.1.1： `mpicc`，`mpic++`等

iii. 示例

- `g++ -pthread -o test.o test.cpp`
- `g++ -fopenmp -o test.o test.cpp`
- `mpic++ -o test.o test.cpp`

注：运行 MPI 命令前需加载 OpenMPI 环境：

- `module load openmpi/4.1.4-mpi-x-gcc9.3.0`

5. 使用任务队列

i. 同步执行，可用于小规模测试。程序执行结束前中断连接会导致程序中止。

- 测试串行程序示例(`test.o`)：

```
yhrun -p thcp1 -n 1 ./test.o
```

- 测试多线程程序示例(`test.o`)，使用 8 个 CPU 核：

```
yhrun -p thcp1 -n 1 -c 8 ./test.o
```

- 测试多进程程序示例(`test.o`)，使用 2 个节点，共 8 个进程，每节点 4 个进程(默认每个进程分配 1 个 CPU 核)：

```
yhrun -p thcp1 -N 2 -n 8 ./test.o
```

ii. 异步执行，常规使用方法。通过提交任务实现，提交任务后可随时退出。

➢ 步骤 1：编写任务脚本。脚本编写及参数设置可参考 SLURM 调度系统：https://slurm.schedmd.com/sbatch.html

- 串行程序脚本示例：

test.sh

```
#!/bin/bash
time yhrun -p thcp1 -n 1 ./test.o &> run.log
```

- 多线程程序脚本示例：

test.sh

```
#!/bin/bash
time yhrun -p thcp1 -n 1 -c 8 test.o &> run.log
```

- 多进程程序脚本示例：

test.sh

```
#!/bin/bash
module load openmpi/4.1.4-mpi-x-gcc9.3.0
time yhrun -p thcp1 -N 2 -n 8 test.o &> run.log
```

➢ 步骤 2：提交任务

- 提交串行程序任务：

```
yhbatch -p thcp1 -n 1 ./test.sh
```

- 提交多线程程序任务：

```
yhbatch -p thcp1 -n 1 ./test.sh
```

- 提交多进程程序任务，使用 2 个节点，共 8 个核，每节点 4 个核：

```
yhbatch -p thcp1 -N 2 -n 8 ./test.sh
```

➢ 步骤 3：查看/删除任务

- 查看任务列表，找到相应 jobid：

```
yhq
```

- 删除任务：

```
yhcancel jobid
```

6. 任务结果查看

- 任务运行结果文件名称默认保存格式为：slurm-jobid.out。

  如：`slurm-418101.out`

- 可以在任务脚本中添加输出重定向，输出到自定义文件中。

  如：`time yhrun -n 1 ./test.o &> run.log`

三、实验题目

实验一：多线程计算正弦值

本实验的输入包含线程数和弧度值。要求：Pthread 并行化实现。

方法：利用正弦函数的泰勒级数展开式计算结果。

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

➢ 编译与运行

- 编译命令：`g++ -pthread -o test.o test.cpp`

- 同步执行命令：`time yhrun -p thcp1 -n 1 -c 8 ./test.o`
- 脚本示例：

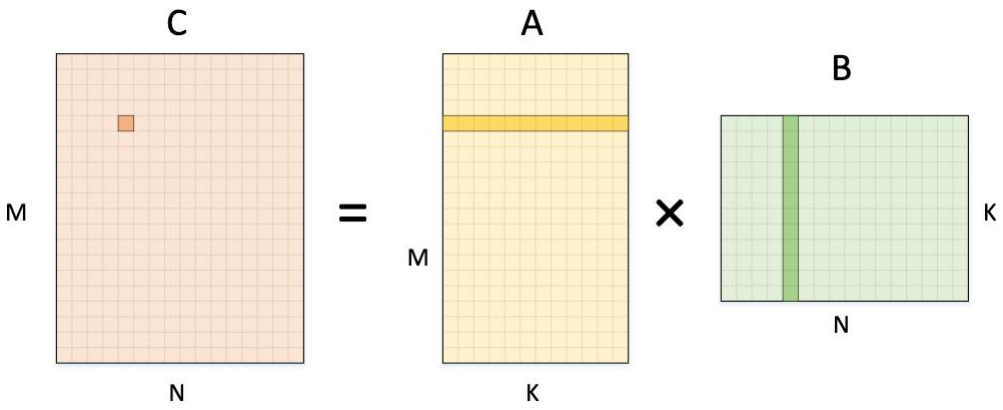
test.sh

```
#!/bin/bash
time yhrun -p thcp1 -n 1 -c 8 test.o &> run.log
```

- 提交任务：`yhbatch -p thcp1 -n 1 ./test.sh`

实验二：多线程计算矩阵乘

气象预报、石油勘探、核子物理等现代科学技术大多依赖计算机的计算模拟，模拟计算的核心是表示状态转移的矩阵计算。另一方面，在人工智能领域，矩阵乘法也是许多核心算法的基础运算。优化 矩阵乘法的性能可以显著提升 AI 模型的训练和推理效率，从而推动 AI 技术的快速发展。通用矩阵乘（简称 GEMM）的一般形式是 C=AB， 其中 A 和 B 涵盖了各自转置的含义。下图是矩阵乘计算中为计算一个输出点所要使用的输入数据。三个矩阵的形状也如图所示



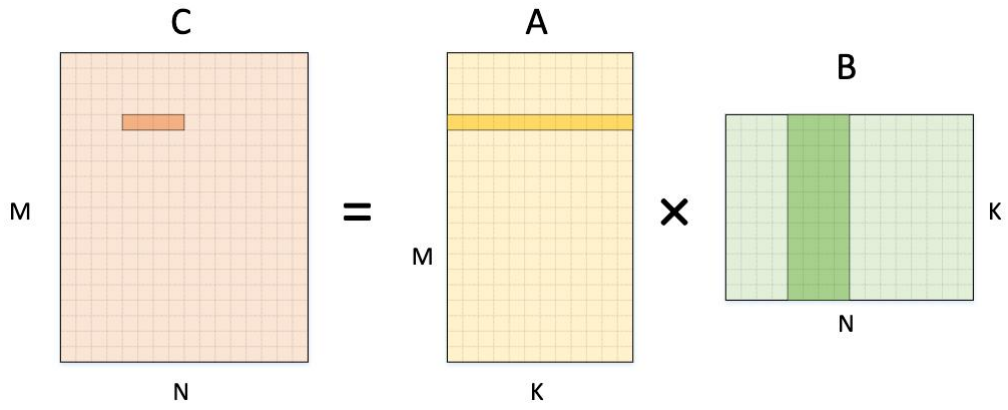
串行代码如下：

```
for (int m = 0; m < M; m++) {
  for (int n = 0; n < N; n++) {
    C[m][n] = 0;
    for (int k = 0; k < K; k++) {
      C[m][n] += A[m][k] * B[k][n];
    }
  }
}
```

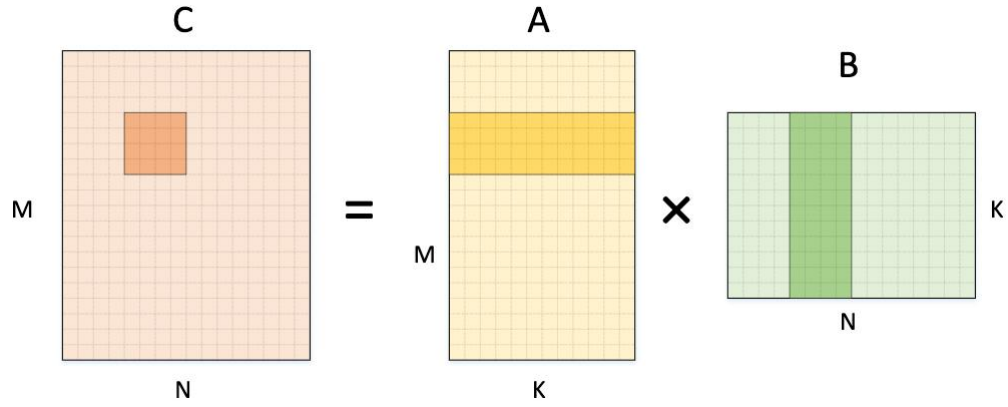
对于 GEMM 性能的优化可以从两个方面入手：

- 软件优化角度：根据计算机体系结构的特点，选择性的调整计算顺序，主要有循环拆分向量化和内存重排等。
- 算法分析角度：根据矩阵乘计算特性，从数学角度优化，例如 Strassen 算法和 Winograd 算法；

例如把输出的计算按列拆分成若干个 1×4 的小块，最内层循环一次计算 4 个值，如下图：



类似的，我们还可以继续分块。把输出的计算按列拆分成若干个 4×4 的小块，最内层循环一次计算 16 个值，如下图：



本实验的输入包含线程数（进程数）、一个数 T，以及 T 组 M，K，N，需要求解不同规模的矩阵乘，并输出最终结果以及所需的时间。要求：Pthread 并行化实现。

i. 串行算法

```
for (int m = 0; m < M; m++) {
  for (int n = 0; n < N; n++) {
    C[m][n] = 0;
    for (int k = 0; k < K; k++) {
      C[m][n] += A[m][k] * B[k][n];
    }
  }}
```

ii. 并行算法

可对每次矩阵相乘进行划分，划分方法可参考课程中的 Jacobi 迭代，将结果矩阵划分成 p（线程数）个子块，每个线程处理一个子块，并行处理矩阵乘，**注意避免假共享。**

➢ 编译与运行

此例中申请 8 个 CPU 核，线程数在程序./test.o 中控制。

- 编译命令：g++ -pthread -o test.o test.cpp
- 同步执行命令：time yhrun -p thcp1 -n 1 -c 8 ./test.o
- 脚本示例：

test.sh

```
#!/bin/bash
time yhrun -p thcp1 -n 1 -c 8 test.o &> run.log
```

- 提交任务：yhbatch -p thcp1 -n 1 ./test.sh

实验三：MPI 多进程计算矩阵乘

本实验针对实验二问题，采用 MPI 编程模型实现矩阵乘计算。

➢ 编译与运行

此例中申请运行 8 个进程，并指定 8 个进程运行到 2 个节点上，默认每个进程分配 1 个 CPU 核。

- 编译命令：**mpic++** -o test.o test.cpp
- 加载 MPI 环境：module load openmpi/4.1.4-mpi-x-gcc9.3.0
- 同步执行命令：time yhrun -p thcp1 -N 2 -n 8 ./test.o
- 脚本示例：

test.sh

```
#!/bin/bash
module load openmpi/4.1.4-mpi-x-gcc9.3.0
time yhrun -p thcp1 -N 2 -n 8 test.o &> run.log
```

- 提交任务：yhbatch -p thcp1 **-N 2 -n 8** ./test.sh

实验四：异构并行计算矩阵乘

本实验针对实验二问题，采用异构并行方式实现矩阵乘计算。

➢ 编译与运行

根据实际环境选择相应的编译与运行命令。