



天津大学
Tianjin University

并行计算实验部分

肖健 杨斌 孙超





实验环境介绍

- 计算集群
 - 国家超级计算天津中心提供
 - 国产飞腾处理器

计算节点配置

CPU	型号	国产自主 FT2000+ @ 2.30GHz 56cores
节点数	5000 个	
内存	128GB	

网络：天河自主高速互联网络 400Gb/s

单核理论性能（双精度）9.2GFlops

单节点理论性能（双精度）588.8GFlops



实验环境介绍

- 编译环境
 - GCC 9.3.0
 - gcc, g++, gfortran等
 - OpenMPI 4.1.4
 - mpicc, mpiCC, mpic++等
- 作业管理系统
 - SLURM 20.11.9



使用软件

- 国家超级计算天津中心提供集成客户端
 - [青索客户端](#)
- 常规使用方式
 - 通过VPN连接内部网络
 - Easy Connect
 - [校内下载](#)
 - 通过远程登陆方式连接集群
 - Win10可直接使用ssh命令
 - Win7需要安装OpenSSH
 - [官方网站](#)
 - 远程文件传输工具
 - FileZilla
 - [官方网站](#)



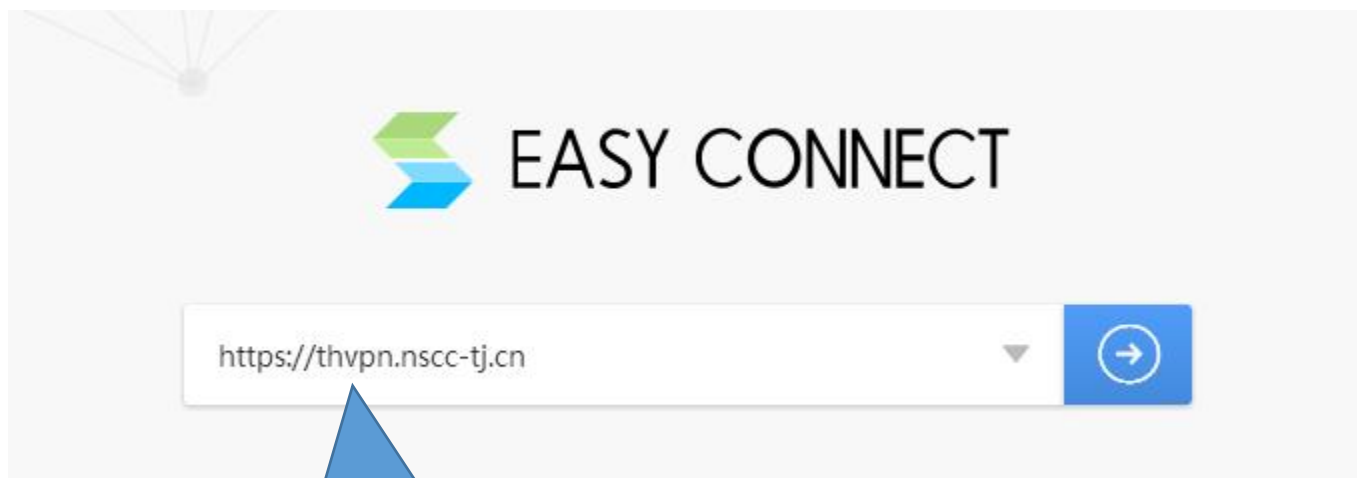
常规流程

- 编写源代码
 - 可在本机（可以是Windows）编写及测试
- 上传源代码
- 登陆集群
 - 编译程序
 - 小规模测试
 - 提交作业
 - 查看运行结果



准备工作

- 登陆VPN
 - 连接国家超级计算天津中心VPN



VPN主机：
<https://thvpn.nsc-ty.cn>



准备工作

- 登陆VPN

NSCC 国家超级计算天津中心 WWW.NSCC-TJ.CN 简体中文 | 诊断

温馨提示

欢迎您使用国家超级计算天津中心客户端!

账号登录

用户名 **用户名: tjucic_course**

密码 **密码: B0%K6vWyeX**

☒ 记住密码 ☐ 自动登录

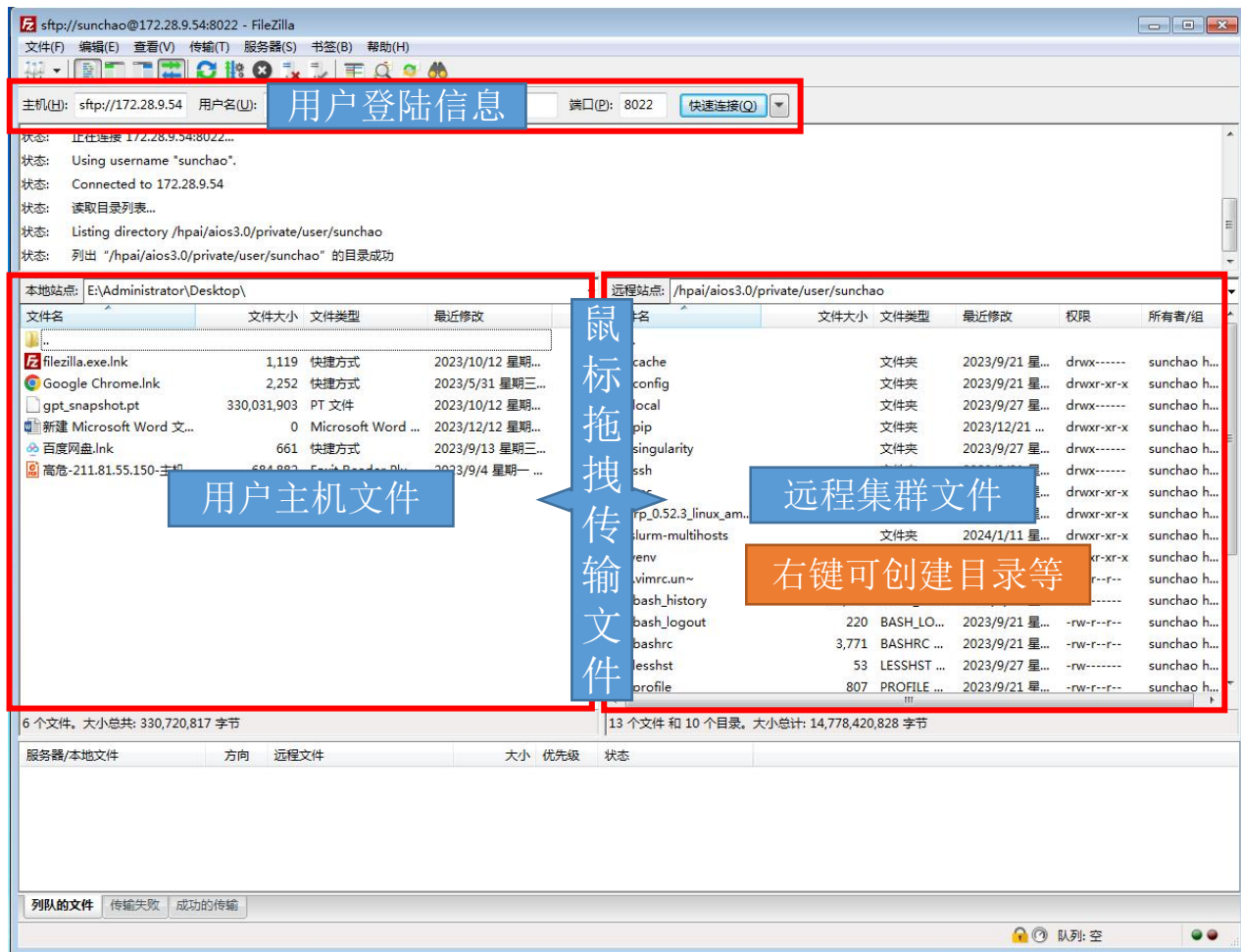
登录

[USB-KEY登录](#) [证书登录](#)



上传源代码

- 文件传输工具





登陆集群

- 登陆集群(命令行)
 - 节点IP: 192.168.10.10
 - 用户名密码 (另行通知)

命令提示符

```
Microsoft Windows [版本 10.0.19042.1466]
(c) Microsoft Corporation。保留所有权利。

C:\Users\administrator>ssh tjucic_course@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
ECDSA key fingerprint is SHA256:KQNjjxdDQZxB3bCSYioQEVGRCsNFuQQapJIBT9RZ5qL.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.10.10' (ECDSA) to the list of known hosts.
tjucic_course@192.168.10.10's password:
*****
* Welcome to NSCC-TJ Supercomputer System.                               *
* For questions or additional information, please contact:                 *
* support@nsc-tj.cn (Hardware) / support@nsc-tj.cn (Software)             *
*****
交互方式执行命令
Disk quotas for grp tjucic_course (id 10271):
  Filesystem      used      quota      limit  grace  files  quota  limit  grace
  /thfs1          268k      512G         1T      -       31 1000000 2000000 -
tjucic_course@ln0: $
```

用户名

节点IP

确认
密钥

密码 (无回显)



常用Linux命令

- `ls` 列出当前目录下文件
 - 如: `ls -al`
- `mkdir` 新建文件夹
- `cd` 切换当前工作目录
 - 如: `cd data/`
- `pwd` 查看当前工作目录
- `rm` 删除文件或目录
- `passwd` 修改登陆密码
- `exit` 退出登陆



编译程序

- 切换到源文件所在目录
 - 如: `cd data/`
- 编译命令
 - 串程序
 - `g++ -o test.o test.cpp`
 - 多线程程序
 - `g++ -pthread -o test.o test.cpp`
 - `g++ -fopenmp -o test.o test.cpp`
 - 多进程程序
 - `module load openmpi`
 - `mpic++ -o test.o test.cpp`



小规模测试

- 串行程序
 - `yhrun -p thcp1 -n 1 ./test.o`
- 多线程程序
 - 使用8个核
 - `yhrun -p thcp1 -n 1 -c 8 ./test.o`
- 多进程程序
 - 共使用2个节点，8个核（每个节点4个核）
 - `yhrun -p thcp1 -N 2 -n 8 ./test.o`
- **注意：不建议使用本机运行方式（`./test.o`）**
- 使用`time`命令可获取程序运行时间
 - 如：`time yhrun -p thcp1 -n 1 ./test.o`



使用队列

- 编写任务脚本（以test.sh为例）

- 脚本示例（串程序）

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

- 脚本示例（多线程程序）

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

- 脚本示例（多进程程序）

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

- 其它脚本编写及参数设置可参考[SLURM调度系统](#)



使用队列

- 提交任务

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

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

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

- 查看任务

```
yhq
```

```
yhcontrol show jobs jobid
```

- 删除任务

```
yhcancel jobid
```



演示

```
tjucic_course@ln0:~$ cd data/
tjucic_course@ln0:~/data$ ls
test.cpp  test.sh
tjucic_course@ln0:~/data$ g++ -pthread -o test.o test.cpp
tjucic_course@ln0:~/data$ yhrun -p thcp1 -n 1 -c 8 ./test.o 10000000 8
16.695311
tjucic_course@ln0:~/data$ time yhrun -p thcp1 -n 1 -c 8 ./test.o 10000000 8
16.695311

real    0m0.619s
user    0m0.301s
sys     0m0.052s
tjucic_course@ln0:~/data$ yhbatch -p thcp1 -n 1 -c 8 ./test.sh
Submitted batch job 418101
tjucic_course@ln0:~/data$ yhq
      JOBID PARTITION    NAME          USER ST       TIME  NODES NODELIST(REASON)
      418101      thcp1  test.sh  tjucic_cours  R       0:01      1 cn1314
tjucic_course@ln0:~/data$ ls
run.log  slurm-418101.out  test.cpp  test.o  test.sh
tjucic_course@ln0:~/data$ cat slurm-418101.out

real    0m10.075s
user    0m0.168s
sys     0m0.020s
tjucic_course@ln0:~/data$ cat run.log
23.603067
tjucic_course@ln0:~/data$ _
```



查看结果

- 作业结果文件
 - `slurm-(jobid).out`
- 上例的结果为:
 - `slurm-418101.out`
- 脚本的最后一行为输出重定向
 - `... &> run.log`



多线程代码示例

```
#include <stdio>
#include <stdlib>
using namespace std;

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("error\n");
    }
    int threadnum=atoi(argv[1]);
    ....
    return 0;
}
```

运行方法: `yhrun -p thcp1 -n 1 -c 8 ./test 8`



进阶：参数化脚本（单节点）

```
#!/bin/bash
if [ "$SLURM_NTASKS" == "" ]; then
    NTASKS=1
else
    NTASKS=$SLURM_NTASKS
fi
if [ "$SLURM_CPUS_PER_TASK" == "" ]; then
    CORES=1
else
    CORES="$SLURM_CPUS_PER_TASK"
fi
PROGRAMS="./test $CORES"
time yhrun -n $NTASKS -c $CORES $PROGRAMS &> run.log
```



实验题目

- 实验一：多线程计算正弦值 (Pthread)
- 正弦函数的泰勒级数展开式：

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

- 注意避免假共享！



实验题目

- 实验二/三/四：计算矩阵乘
 - 串行计算伪代码
 - 输入
 - 两个二维矩阵
 - 进行矩阵相乘
 - 输出
 - 结果矩阵
 - 并行思路
 - 按结果矩阵进行划分

```
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];  
        }  
    }  
}
```



实验题目

- 实验二：多线程（Pthread）计算矩阵乘
 - 按数据块划分计算量
- 实验三：多进程（MPI）计算矩阵乘
 - 参考Jacobi迭代的划分
 - 每个进程处理一块结果
 - 注意消息传递
- 实验四：异构并行计算矩阵乘
 - 具体实验环境另行通知
 -



Thank you !

