

天津大学超算中心
High Performance Computing Center

并行计算实验部分
2018级

孙超 sch@tju.edu.cn



2018-9-29

实验简介

- 实验课时计划安排
- 实验内容安排
- 实验要求及报告
- 实验环境介绍

实验课时计划

- 实验一

- 10月12日下午四节

- 实验二

- 10月19日下午四节

- 实验三

- 10月26日下午四节
- 11月9日下午后两节

实验内容安排

- 实验一(50%) (25%)
 - Pthread程序设计
- 实验二(50%) (25%)
 - MPI程序设计
- 自选实验（博士）(50%)
 - 结合自己科研工作

实验报告

- 实验完成后，要提交实验报告（电子版）
 - 提交时间为**11月16日之前**
 - **每次一个**实验报告
 - 在指定网站提交
 - 最后统一提交
- 实验报告内容
 - 实验题目，作者，时间
 - 实验内容，原理，程序流程图，实现方法
 - 结果（数据图表、效果图等），理论性能分析以及对实际结果的分析
 - 总结展望（心得体会等）

实验报告提交

● 实验报告提交格式

- 实验报告使用Word/PDF文档格式
- 源代码使用zip格式提交，不用rar
 - 需要程序代码文件，任务运行PBS脚本
 - 源代码中不要打包输入数据
- 提交网站：**<http://211.81.50.36>**
 - 可修改两次
 - 两次实验依次提交

实验要求

- 必须在集群上进行实验
- 各文件一定要在data文件夹下
 - 程序源代码
 - 运行脚本
 - 可执行程序

实验环境介绍

- 操作系统
 - CentOS
- 编译环境
 - GCC 4.4.7
 - gcc, g++, gfortran, gcj等
 - Intel 13.0
 - icc, icpc, ifort等
 - OpenMPI 1.8.1
 - mpicc, mpiCC等
- 作业管理系统
 - Torque 5.1.3

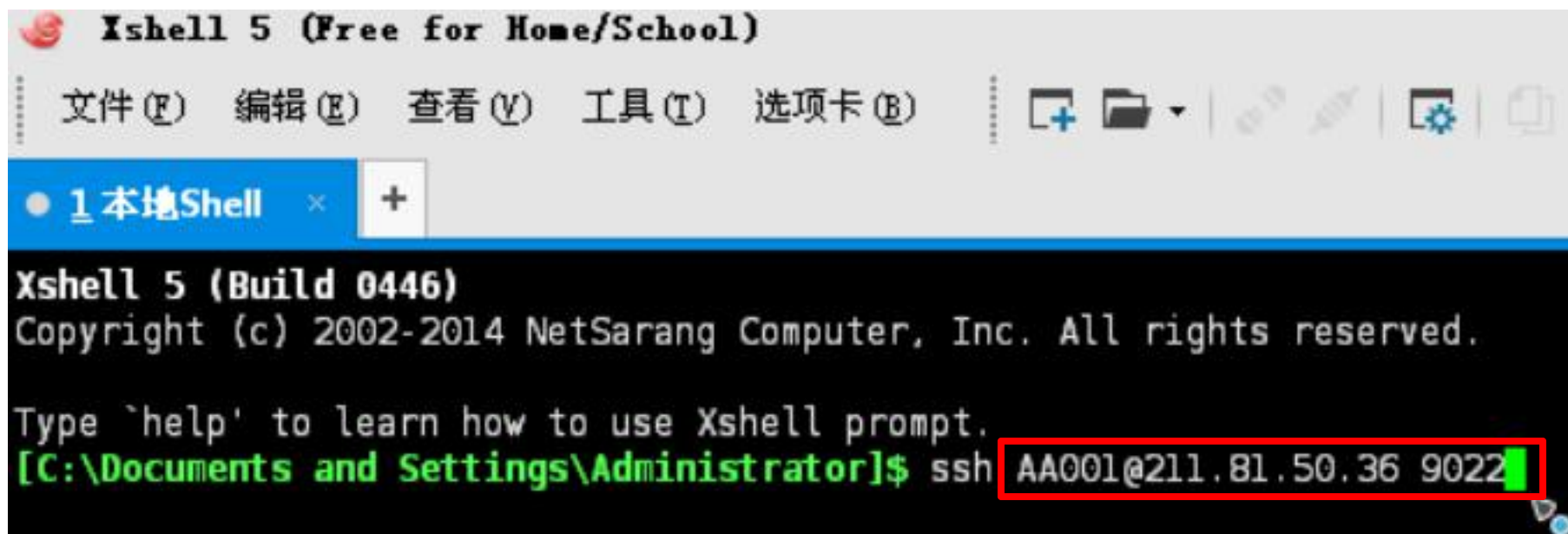
使用方式

- 通过远程登陆方式连接集群
 - 不支持图形界面
 - XShell
 - SSH Secure Shell Client
 - Putty
- 文件传输客户端软件
 - XFTP
 - Secure File Transfer Client
 - WinSCP

使用流程（演示）

- 登陆集群
- 编写程序
- 编译程序
- 编写PBS脚本
- 提交作业
- 查看运行状态
- 查看结果

登陆集群



```
Xshell 5 (Free for Home/School)

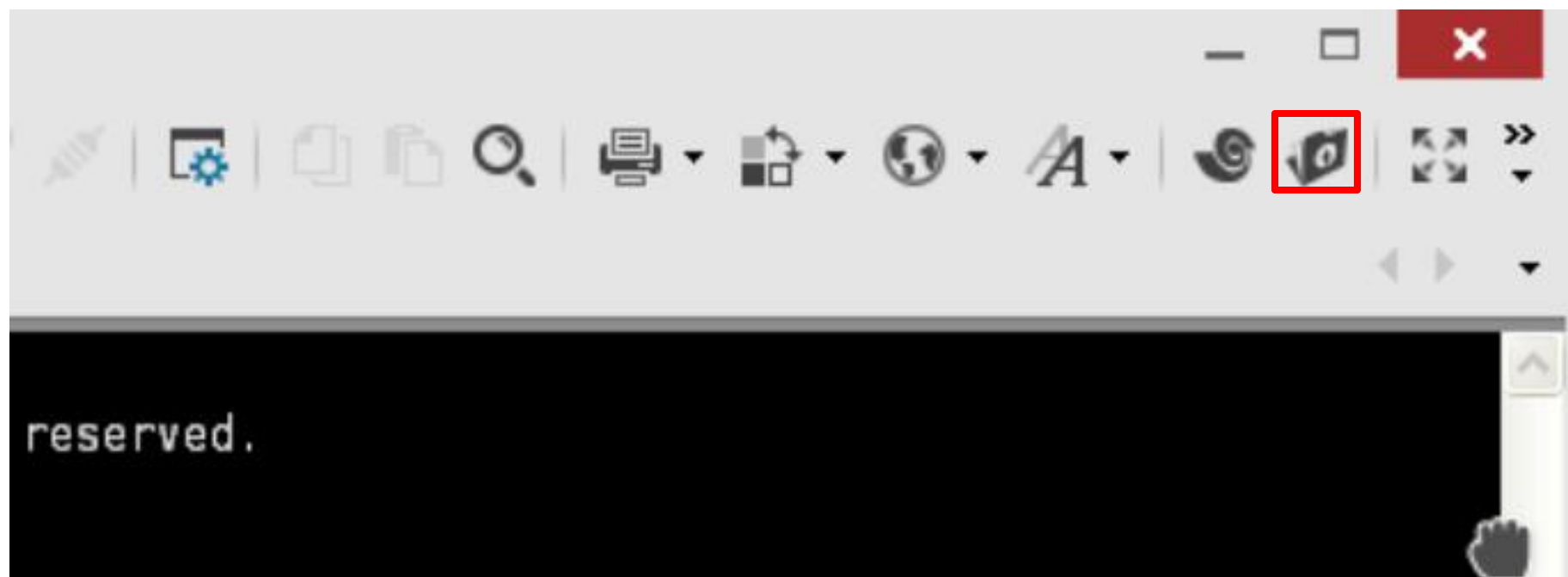
文件(F)  编辑(E)  查看(V)  工具(T)  选项卡(B)

● 本地Shell x +

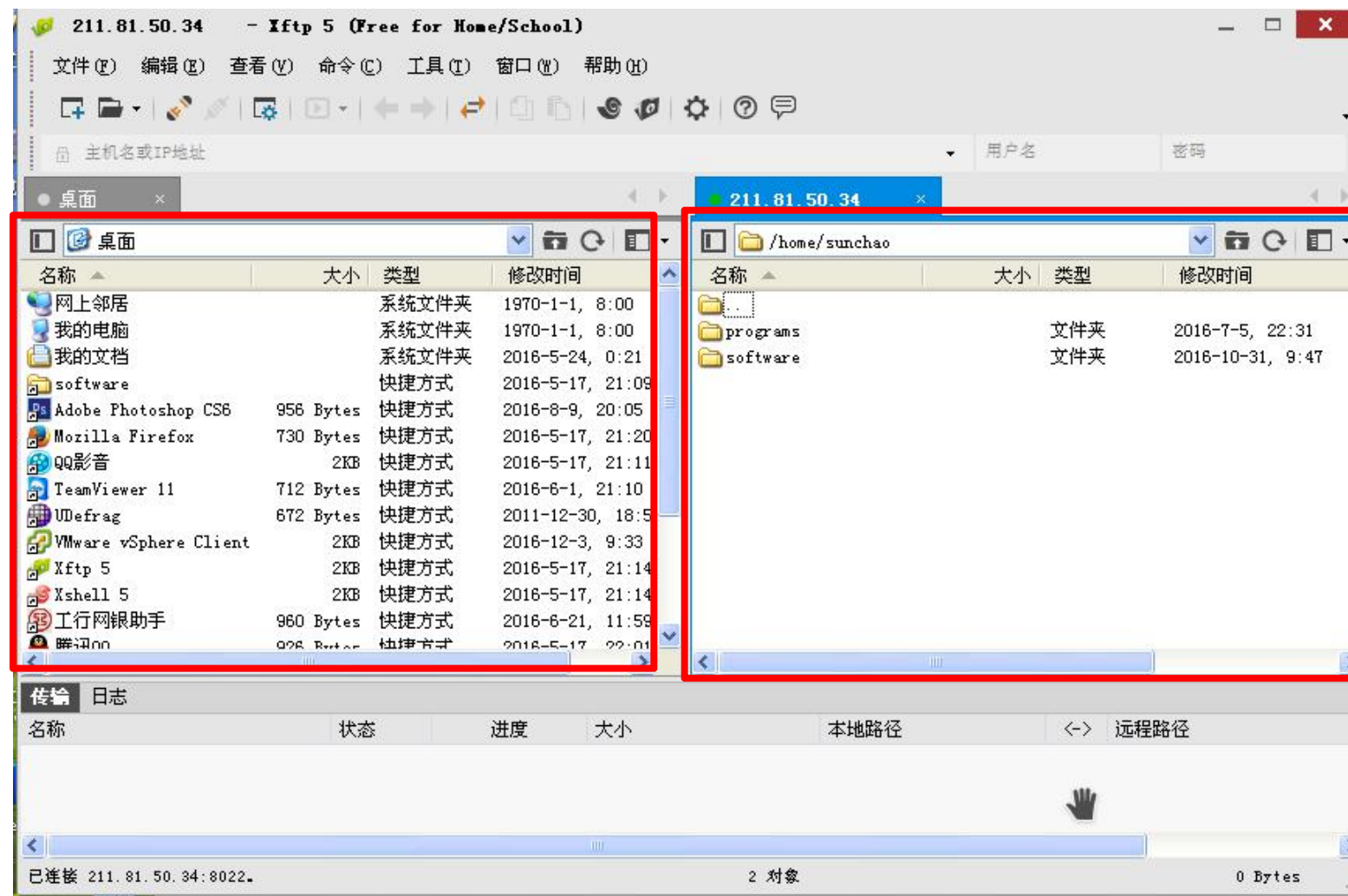
Xshell 5 (Build 0446)
Copyright (c) 2002-2014 NetSarang Computer, Inc. All rights reserved.
Type 'help' to learn how to use Xshell prompt.
[C:\Documents and Settings\Administrator]$ ssh AA001@211.81.50.36 9022
```

- Host Name: 211.81.50.36
- User Name: 会放到glearning网站上
- Port: 9022

文件传输



文件传输



常用Linux命令

- `ls` 列出当前目录下文件
 - 如: `ls -al`
- `cd` 切换工作目录
 - 如: `cd program/`
- `pwd` 查看当前目录
- `rm` 删除文件或目录
- `mv` 文件或目录移动或重命名
- `passwd` 修改登陆密码
- `exit` 退出登陆

一般流程

- 先在本机（可以是Windows）编写好串行程序代码test.cpp，能够正确运行
- 通过文件传输界面，将文件传到集群上，最好在data目录下。
- 能过命令行界面切换到相应目录，上一步的data
 - cd data
- 编译程序
 - gcc -o test test.cpp
- 执行test程序
 - ./test

编译命令

- 示例(多线程与多线程):
 - `icpc -pthread -o test.o test.cpp`
 - `icpc -openmp -o test.o test.cpp`
 - `mpic++ -i_dynamic -o test.o test.cpp`

使用队列

- PBS脚本示例（串程序序）

```
#!/bin/bash
```

```
#PBS -N test
```

```
#PBS -l nodes=1:ppn=1
```

```
#PBS -j oe
```

```
cd $PBS_O_WORKDIR
```

```
./test（如果程序需要参数，可直接写上）
```

使用队列

- PBS脚本示例（MPI并行）

```
#!/bin/bash
```

```
#PBS -N test
```

```
#PBS -l nodes=2:ppn=4
```

```
#PBS -j oe
```

```
cd $PBS_O_WORKDIR
```

```
procs=(cat $PBS_NODEFILE | wc -l)
```

```
mpirun -np $procs -machinefile $PBS_NODEFILE
```

```
./test（如果程序需要参数，可添加）
```

使用队列

● PBS脚本的编写

- 第一行: `#!/bin/bash`
- `#PBS -N` 作业名字(test)
- `#PBS -l` 申请资源(nodes=1:ppn=8)
- `#PBS -j oe`
- `cd $PBS_O_WORKDIR`
- `procs=$(cat $PBS_NODEFILE | wc -l)`
- 运行程序
 - `mpirun -np $procs -machinefile $PBS_NODEFILE ./test`
 - 不要使用后台执行方式

使用队列

- 提交作业
 - `qsub test.pbs`
- 查看作业
 - `qstat -R`
 - `qstat -f jobid`
- 删除作业
 - `qdel jobid`

演示

```

TT01@node63:~> cd data/
TT01@node63:~/data> ls
test.pbs  test.pbs~
TT01@node63:~/data> qsub test.pbs
35.node63
TT01@node63:~/data> qstat

```

Job id	Name	User	Time Use	S	Queue
35.node63	test	TT01	0	R	qstudent

```

TT01@node63:~/data> qstat -R

node63:

```

Job ID	FAST	PFS	Username	Queue	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
35.node63	--	--	TT01	qstudent	1	4	--	01:00	R	--

```

TT01@node63:~/data> qdel 35

```

查看结果

- 作业结果文件
 - (name).o(id)
- 上例的结果为：
 - test.o35
- 可以PBS脚本的最后一行加上重定向
 - ./test >& run.log

多线程代码示例

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

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

运行方法: **./test 8**

实验题目

● 荒野求生(多线程,多进程)

● 问题描述

- 矩形空间内有若干个探险小队
- 每个探险小队有初始的位置和速度
- 碰撞到空间边缘则会转弯
- 相遇会发生冲突,冲突后速度最慢小队会生存下来
- 求若干时间后存活的小队的位置和速度大小以及方向

● 矩形区域分块

- 每一次迭代结束后同步

Thank you !

