

中山大学高性能计算公共平台（珠海校区）使用指南

一、超算使用概述

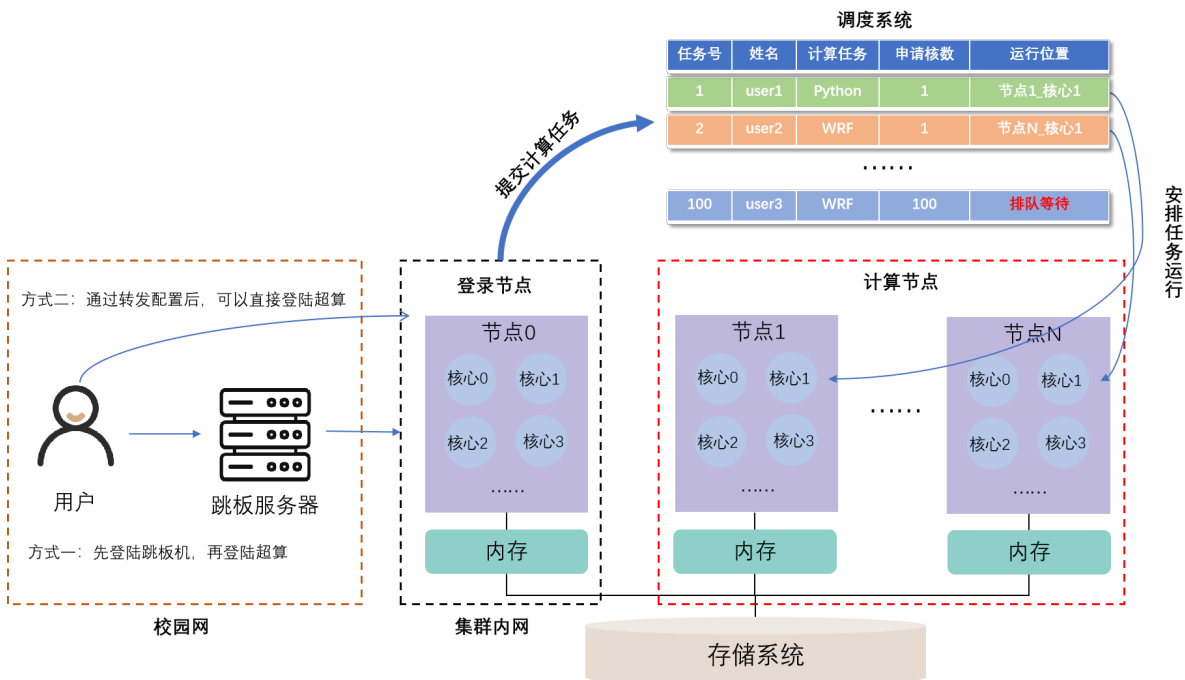
1.1 超算组成

超算主要由四部分组成：跳板服务器、登陆服务器、计算服务器、调度系统

- 跳板服务器。一台既连接了校园网又连接了超算内网的服务器。用户通过该机器可以从校园网中访问处于内网中的超算集群。**IP地址为：172.16.108.134，端口为22。**
- 登陆服务器。**专门给用户远程登陆使用的服务器。IP地址为：192.168.10.15，端口为22。**用户凭集群及密码登入服务器，然后在服务器上可进行**文件上传下载、文件编辑、程序编译、软件安装、计算任务提交**等操作，**但不能直接运行计算任务**，否则会导致机器卡顿，影响其他用户登陆及使用。
- 计算服务器。**专门用来运行计算任务的服务器。**计算服务器配置：**Intel(R) Xeon(R) Gold 6348 CPU \56核\512G内存\8块Nvidia A800 80G显存，总共7台**
- 调度系统。所有计算服务器由调度系统分配管理。用户首先向调度系统申请计算资源，然后再由调度系统将计算任务投放到分配的计算服务器上运行。

1.2 使用步骤

- 用户先登陆跳板机。**为了保证安全，目前超算集群放置在内网中，只有一台位于校园网中的跳板机可以访问超算。因此，为了使用超算，我们需要先登陆到跳板机。
- 再通过跳板机登陆到登陆节点。**登陆到跳板机后，我们可以通过ssh等方式登陆到登陆节点，进行任务脚本的编写和提交。
- 编写计算任务提交脚本。**这个脚本包含了向调度系统申请计算资源的指令，以及定义程序运行命令的参数。通过编写这个脚本，我们可以灵活地配置计算任务的提交方式，以满足不同的计算需求和优化要求。
- 执行任务提交脚本。**将计算任务投放到计算服务器上运行。
- 执行命令查看程序运行状态。**



第一步和第二步，经过一定配置后，可以融合为一步，用户进行一次操作，即可直接访问处于内网的登陆节点，具体设置请参考2.2

二、集群登陆

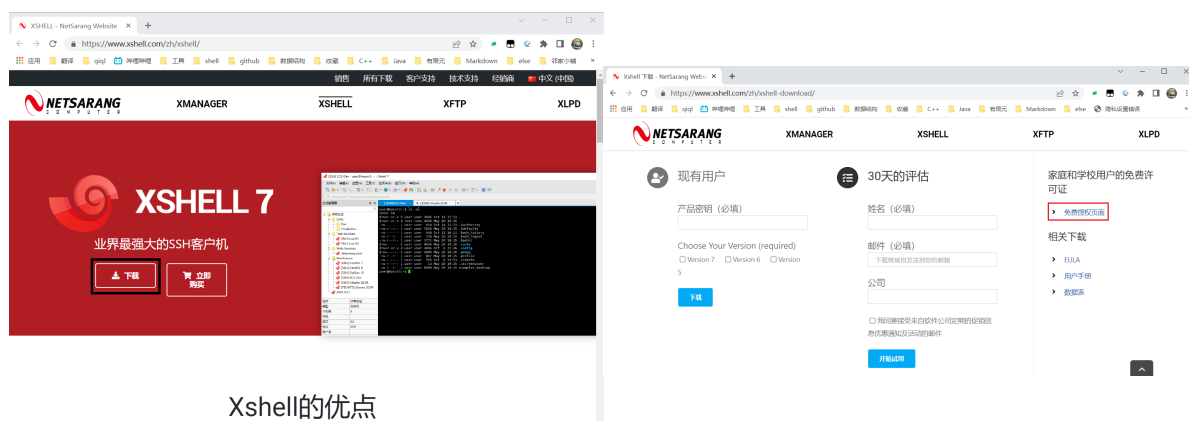
2.1 Windows系统下的集群登陆

2.1.1 集群登陆软件Xshell下载与安装

什么是 Xshell ?

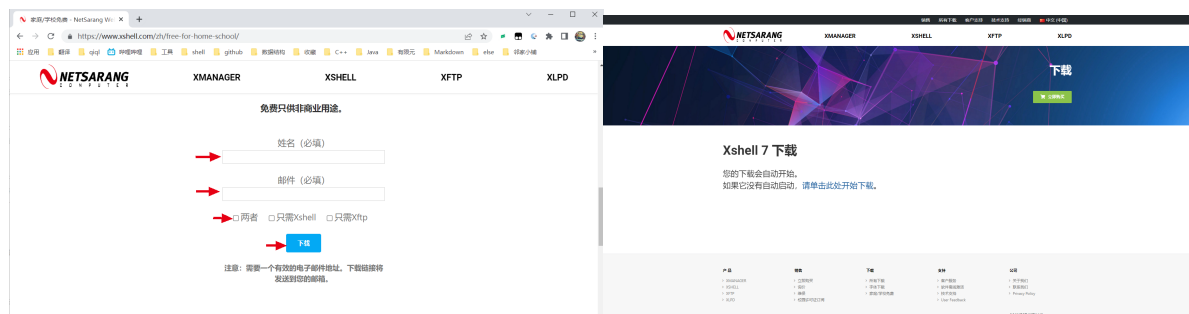
Xshell 是一个用来在 Windows 连接远程服务器的终端工具，Xshell 使用方便且免费。

进入XShell的中文官网:<https://www.xshell.com/zh/xshell/> 进入如下界面，目前版本为 Xshell7，点击下载，点击免费授权页面



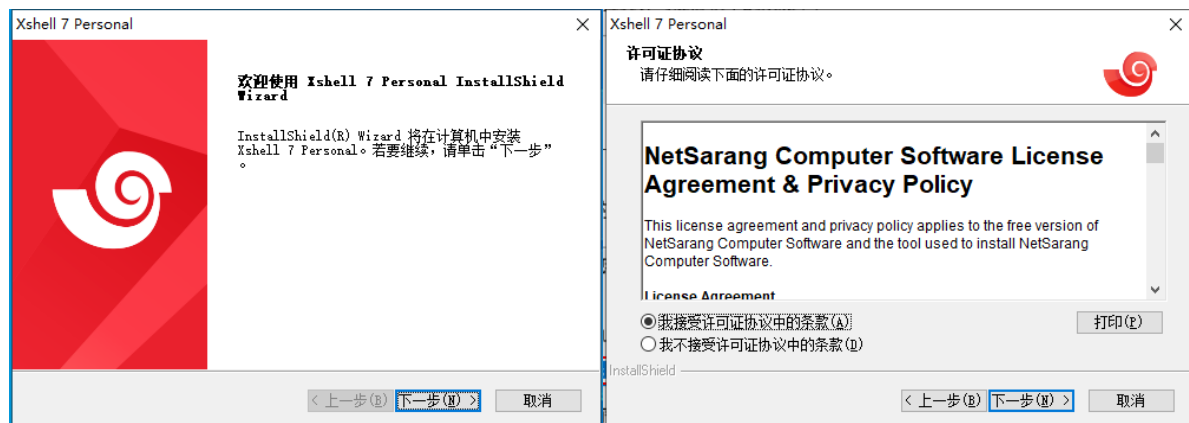
Xshell的优点

进入免费授权页面后，填写姓名和邮件，勾选两者（Xftp 也需要使用），点击下载。注意：需要一个有效的邮件地址，下载链接将发送到邮箱。然后点击邮箱中的下载链接，浏览器将自动下载 Xshell 和 Xftp 的安装包。

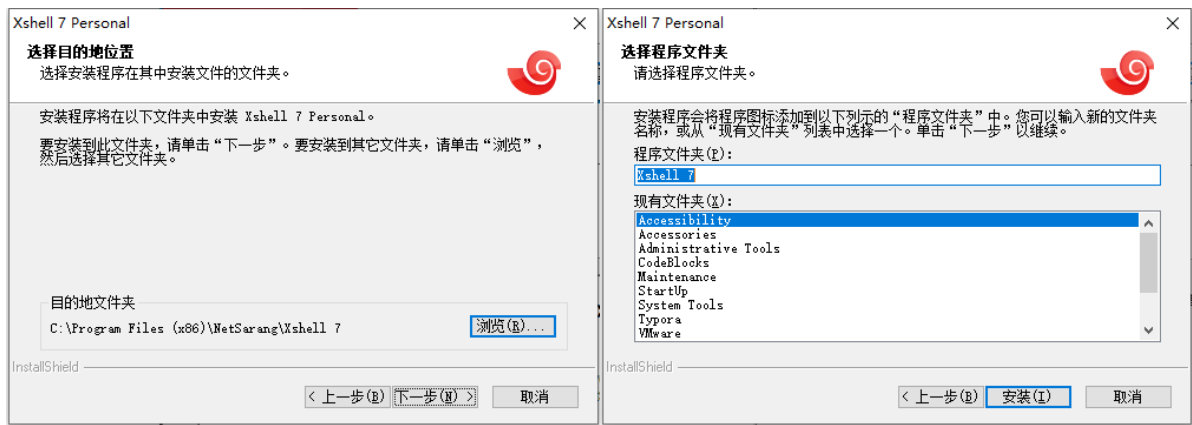


Windows 下安装 Xshell

打开 Xshell7 安装包，点击下一步，勾选我接受，点击下一步



可以选择安装路径或直接点击下一步，点击安装

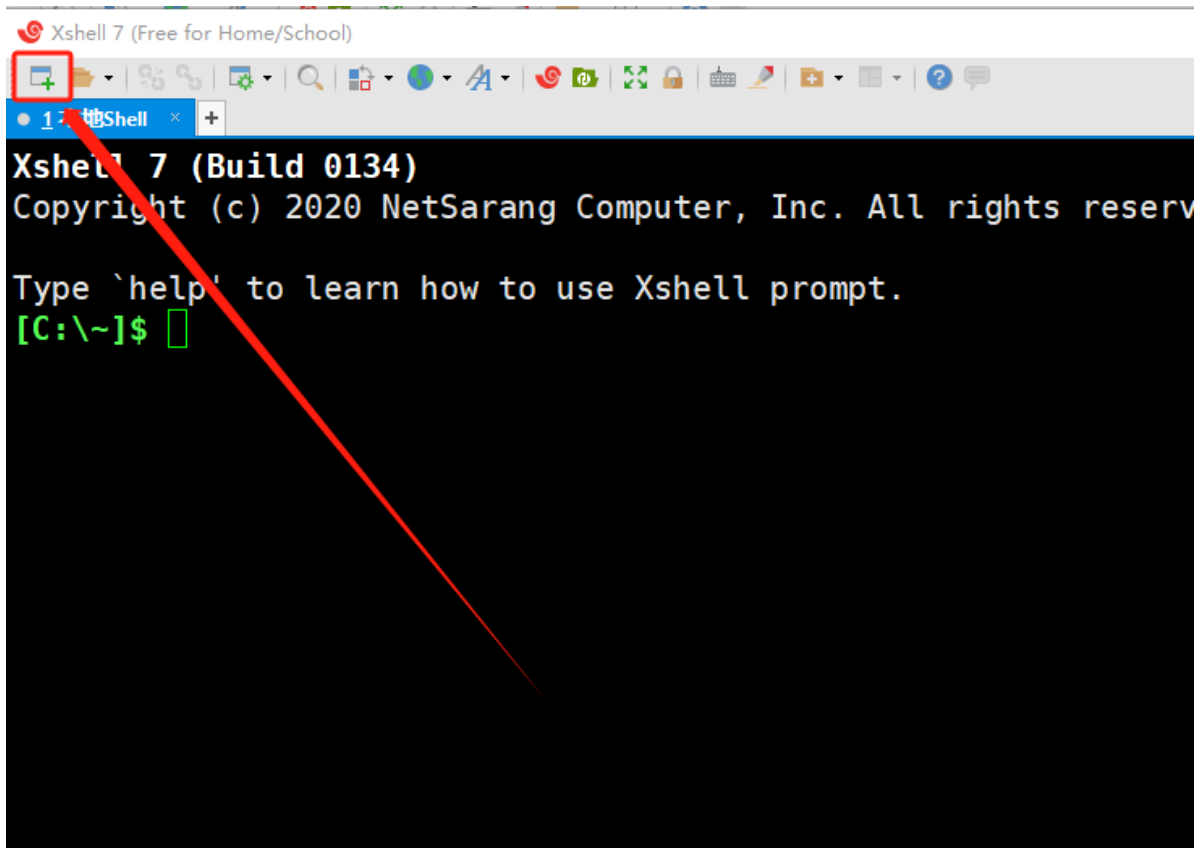


安装完成

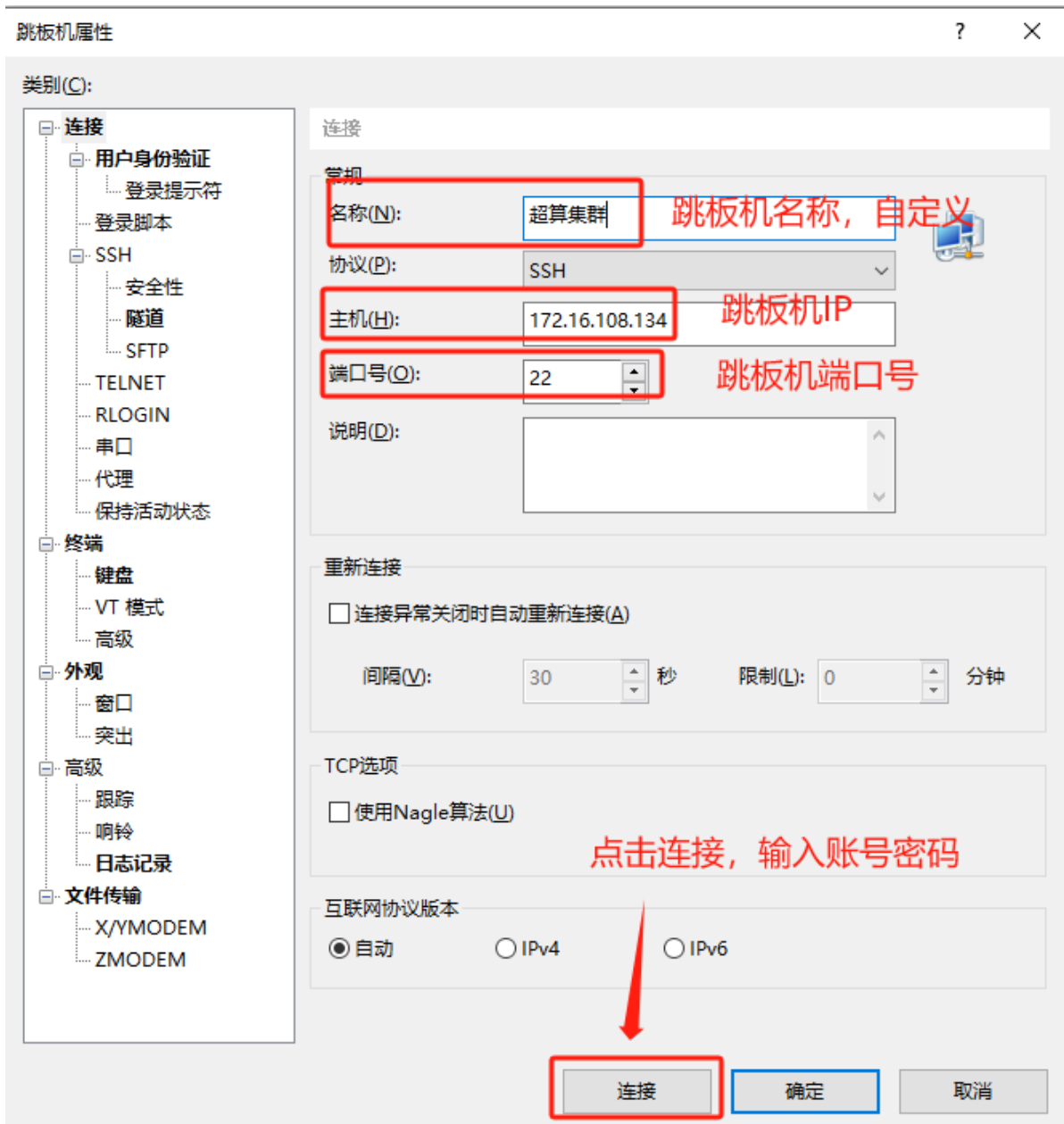


2.1.2 使用Xshell登陆集群

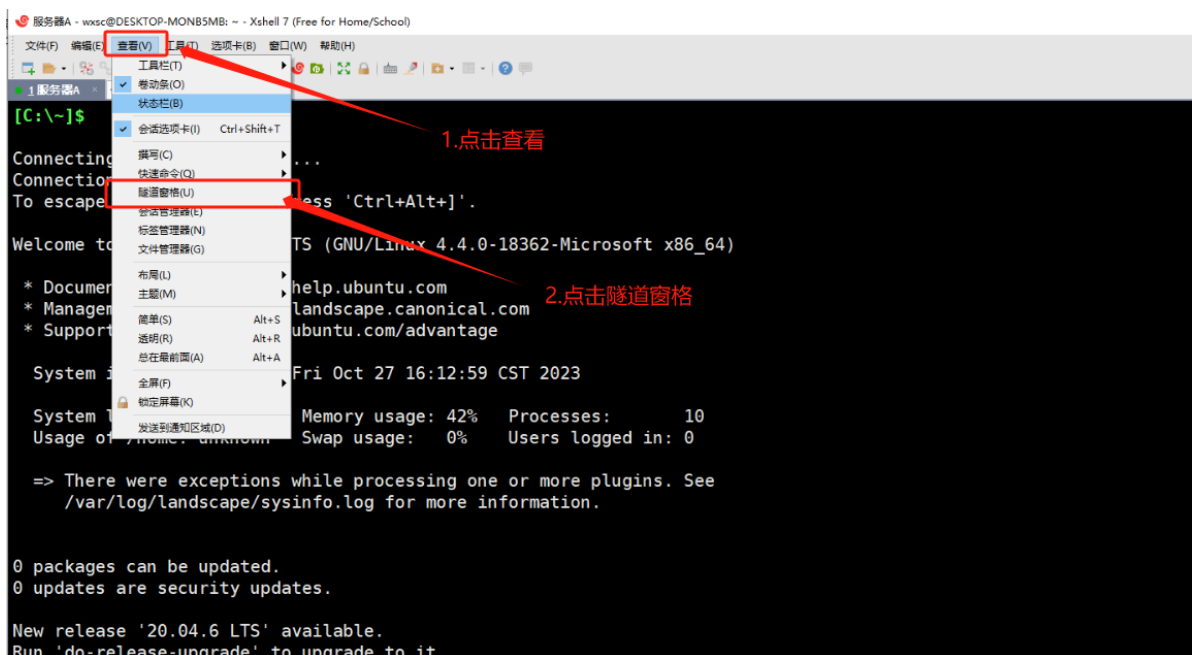
1. 打开xshell，点击左上角“新建”按钮



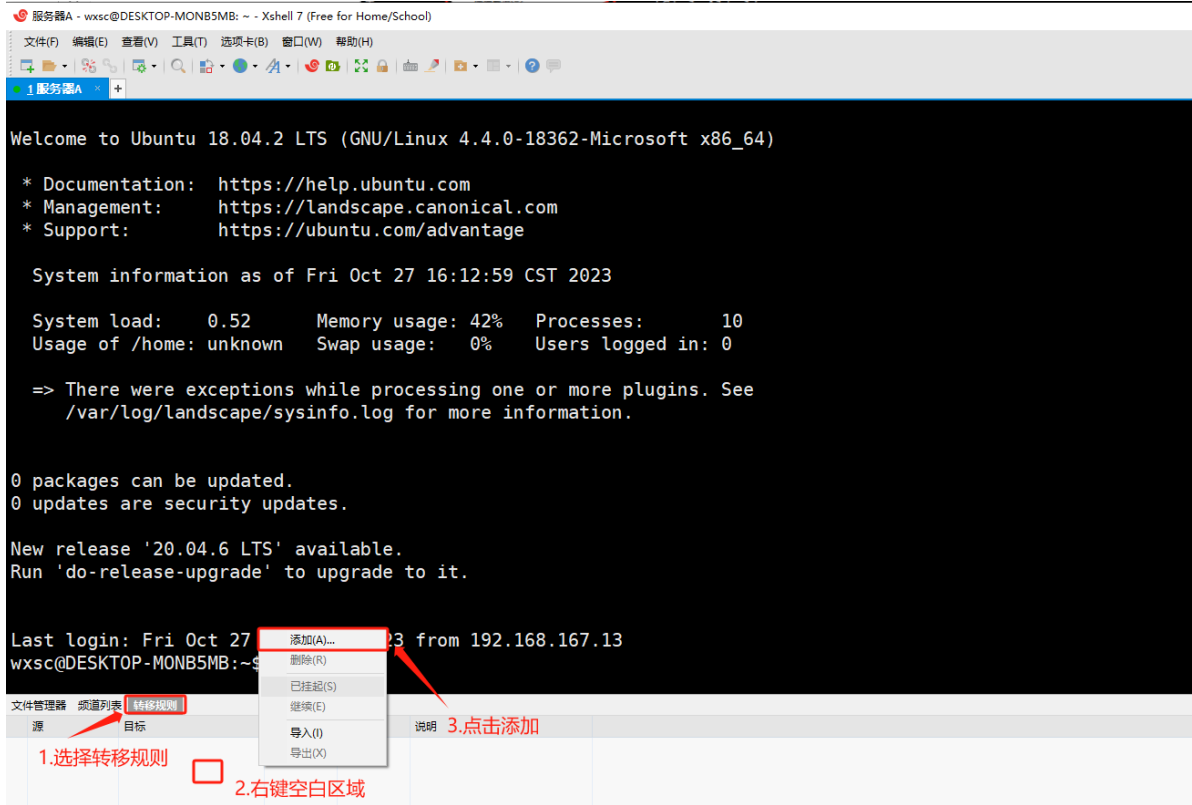
2. 输入跳板机名称, IP: 172.16.108.134, 端口: 22, 名称可以任意, 然后点击“连接”



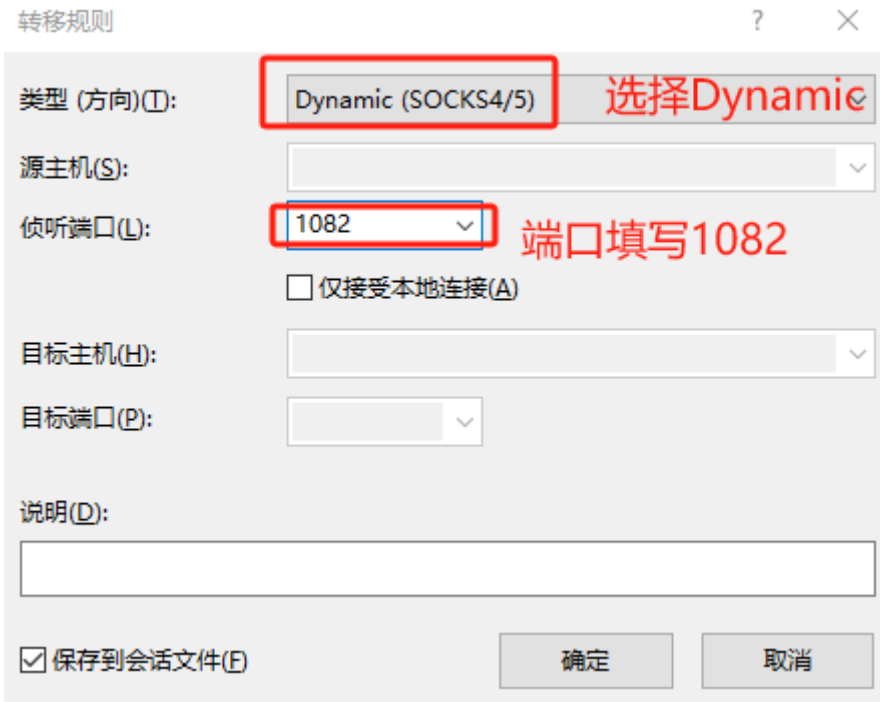
3. 登陆成功之后, 点击“查看”, 然后点击“隧道窗格”, 创建隧道, 步骤如下图



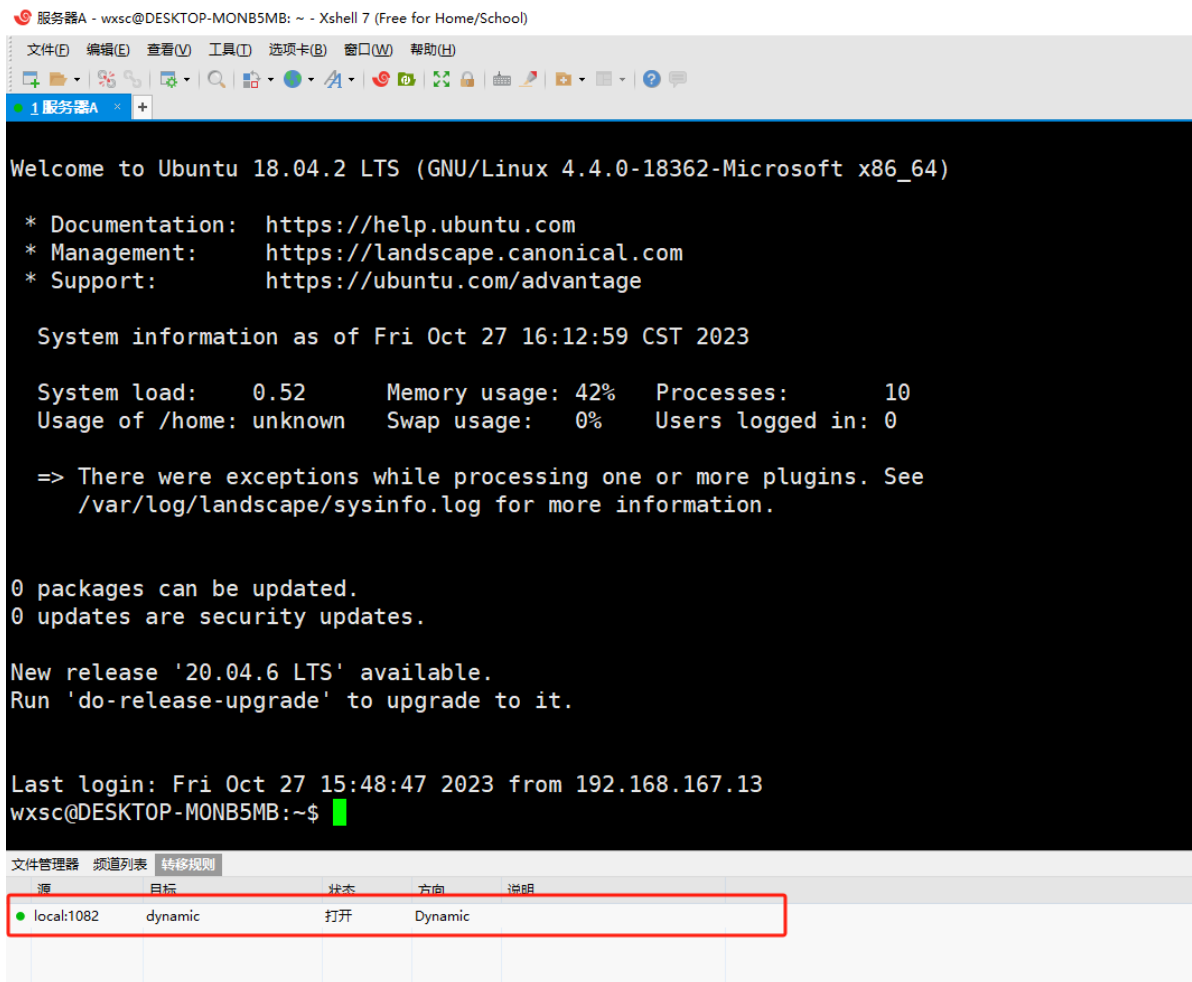
此时xshell下方会出现窗格, 点击“转移规则”, 右键空白区域, 点击“添加”



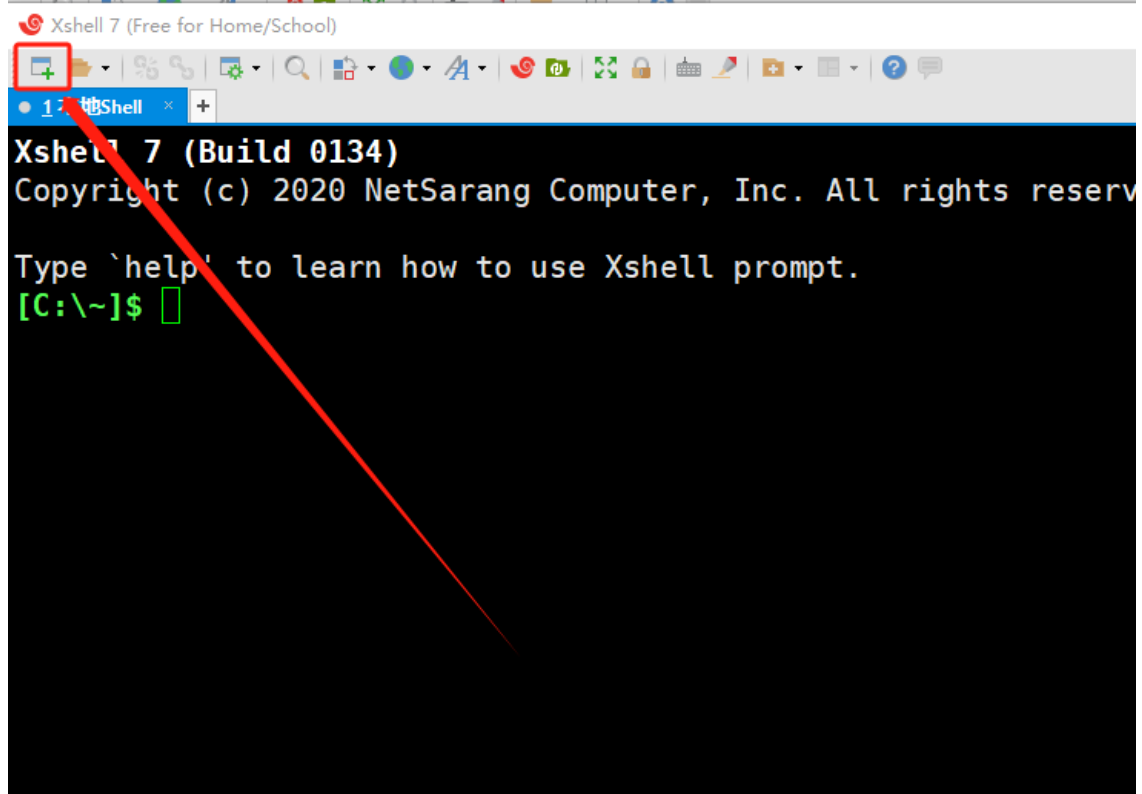
转移规则中，类型选择“Dynamic (SOCKS4/5)”，侦听端口填写 1082，最后点击“确定”



转移规则中，显示出规则并颜色显示为绿色，表示添加并使用成功，跳板机的隧道建立成功



4. 进行登陆节点的配置，进入Xshell首页，点击左上角“新建”按钮



输入登陆节点的名称，IP: 192.168.10.15，端口: 22，名称可以任意

类别(C):

- 连接
 - 用户身份验证
 - 登录提示符
 - 登录脚本
 - SSH
 - 安全性
 - 隧道
 - SFTP
 - TELNET
 - RLOGIN
 - 串口
 - 代理
 - 保持活动状态
- 终端
 - 键盘
 - VT 模式
 - 高级
- 外观
 - 窗口
 - 突出
- 高级
 - 跟踪
 - 响铃
 - 日志记录
- 文件传输
 - X/YMODEM
 - ZMODEM

连接

常规

名称(N): 登录节点 登陆节点名称, 自定义

协议(P): SSH

主机(H): 192.168.10.15 登陆节点的IP

端口号(O): 22 登陆节点的端口号

说明(D):

重新连接

☐ 连接异常关闭时自动重新连接(A)

间隔(V): 30 秒 限制(L): 0 分钟

TCP选项

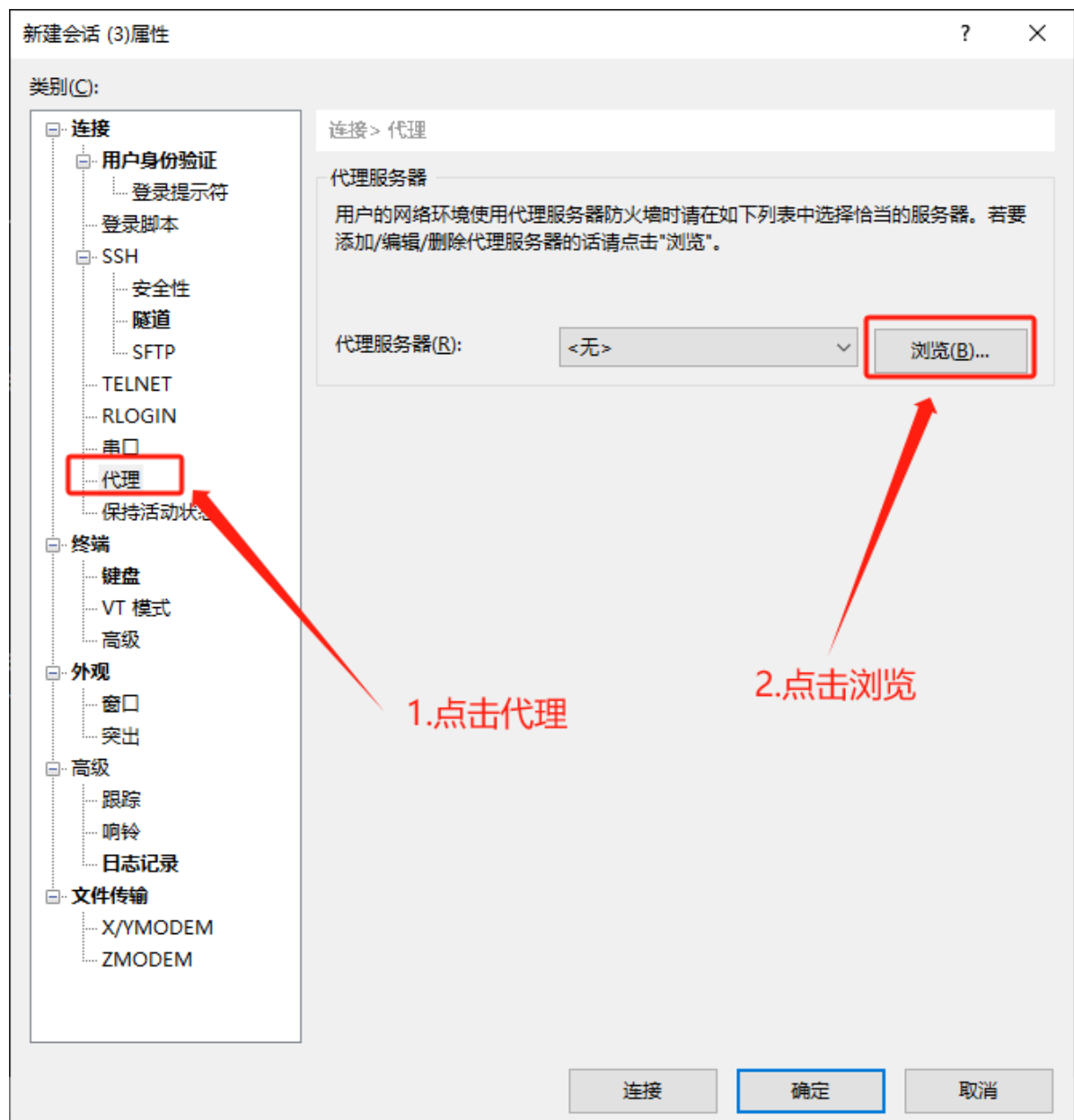
☐ 使用Nagle算法(U)

互联网协议版本

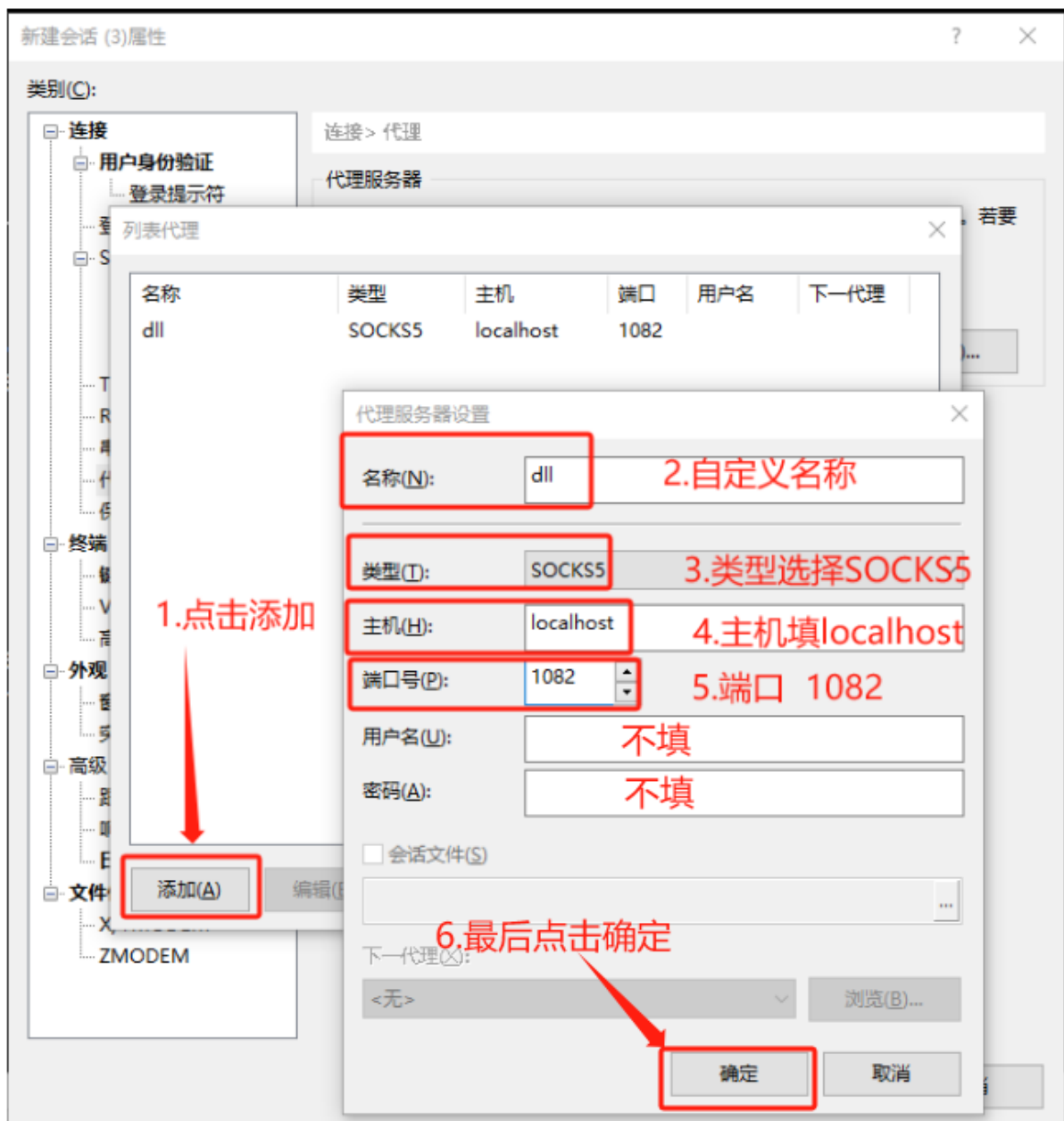
☒ 自动 ☐ IPv4 ☐ IPv6

连接 确定 取消

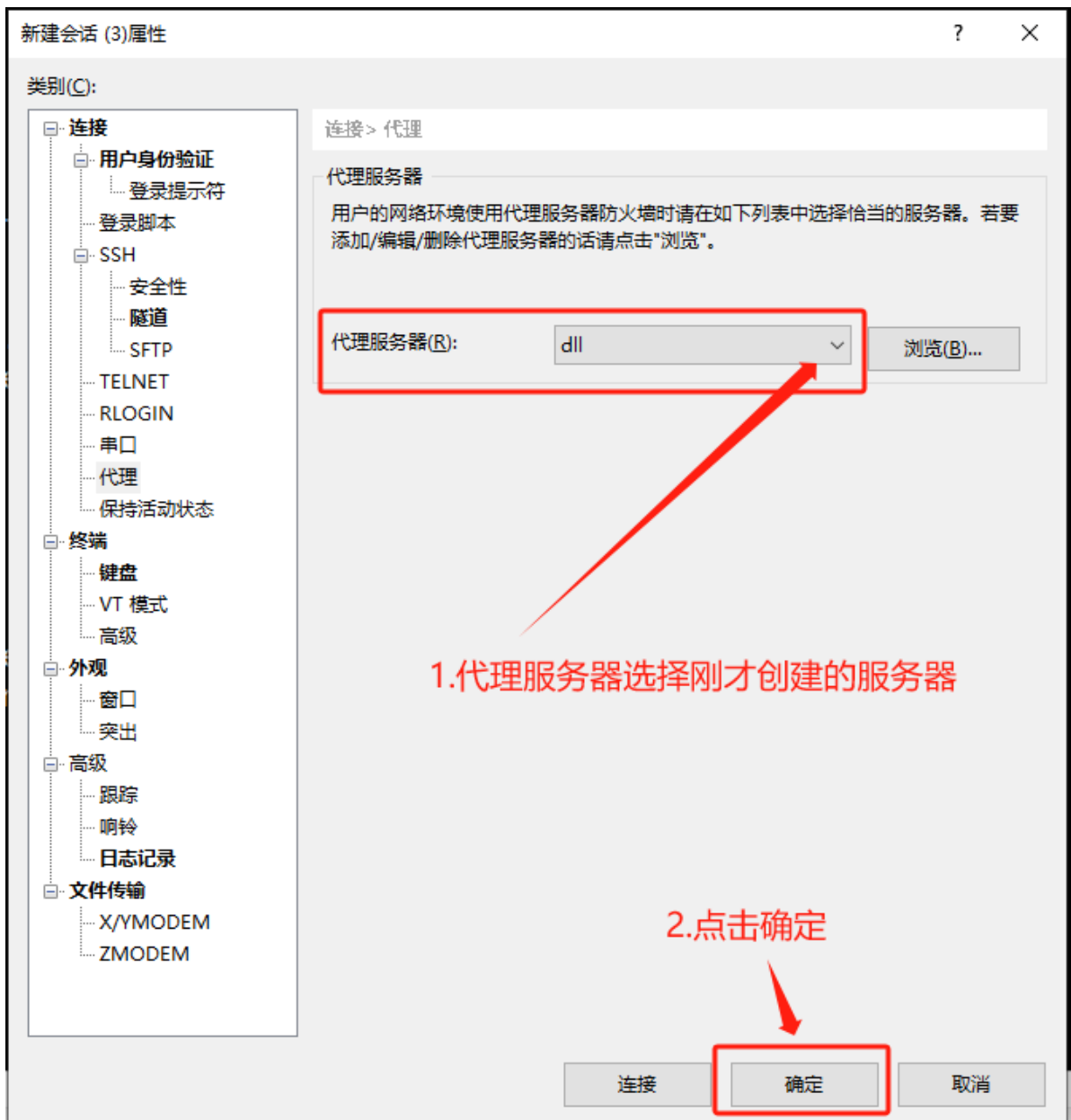
点击左侧“代理”，进入代理配置界面，点击代理服务器右侧的“浏览”



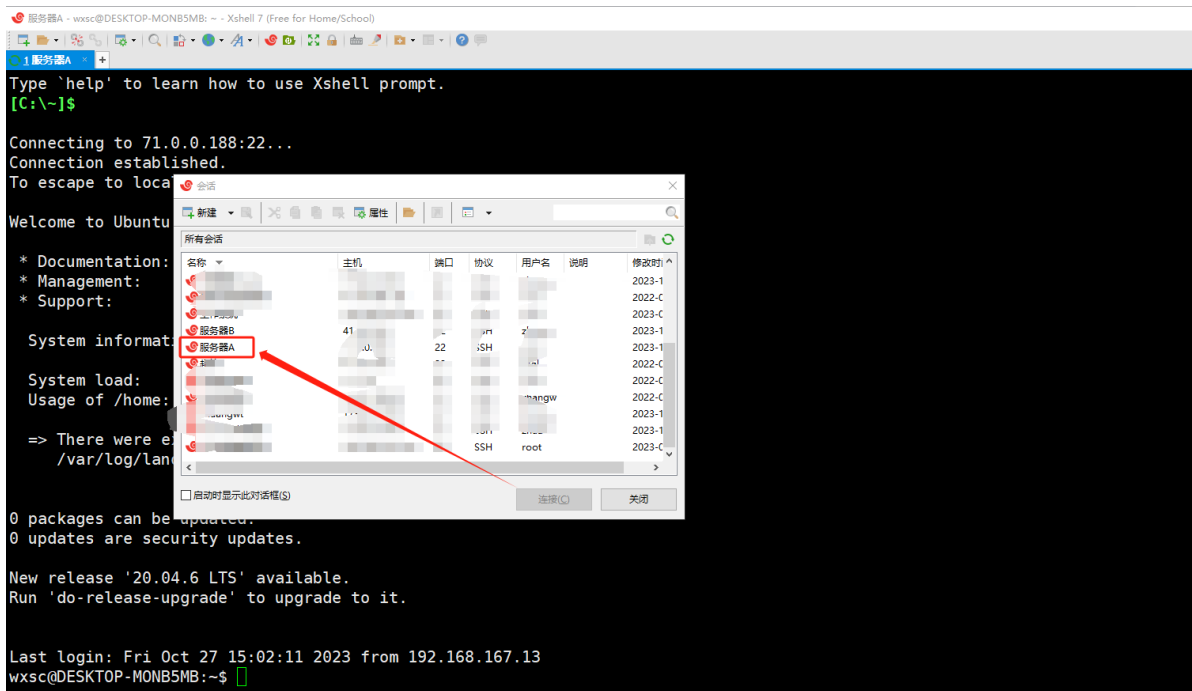
点击“添加”，代理服务器的名称自行定义，类型选择 socks5，主机填写 localhost，端口号 1082，用户名和密码不需要填写，最后点击“确定”

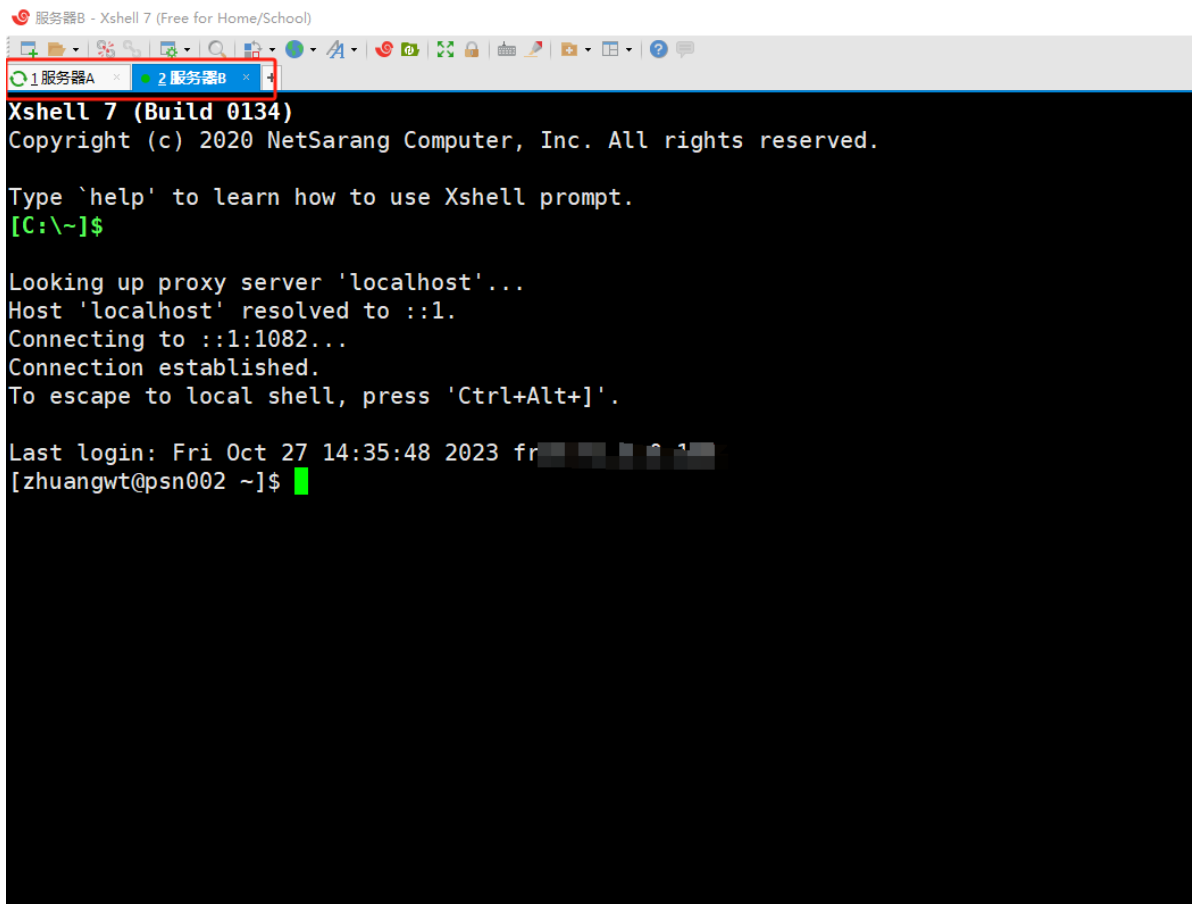


在代理配置界面，将代理服务器选择为创建的代理服务器，并点击“确定”，Xshell配置完成



4.先选择跳板机（名称根据自定义名称进行选择，我这里为服务器A），选择后进行连接，登陆成功





2.2 Linux系统下的集群登陆

在linux系统下登陆集群，我们也可以实现一步登陆，即采用 ssh 的 -J 选项，来指定跳板机的地址和用户名，然后通过跳板机访问登陆节点。

命令格式为：

```
ssh -J username1@172.16.108.134 username2@192.168.10.15
```

其中 172.16.108.134 为跳板机的主机地址，username1 为登陆跳板机时使用的用户名。

192.168.10.15 为登陆服务器的主机地址，username2 为登陆服务器时使用的用户名。

3.1 管理节点（manage）联网

3.1.1 集群文件传输软件Xftp安装

只有跳板机才与校园网连通，可以上网，所以管理节点需要通过跳板机作为代理才能上网，可使用以下命令，声明代理即可上网：

```
export http_proxy=http://192.168.10.22:3128
export https_proxy=http://192.168.10.22:3128
```

其中192.168.10.22 为跳板机节点在内网的IP，3128为代理端口

三、集群文件上传与下载

3.1 Windows系统下的文件上传与下载

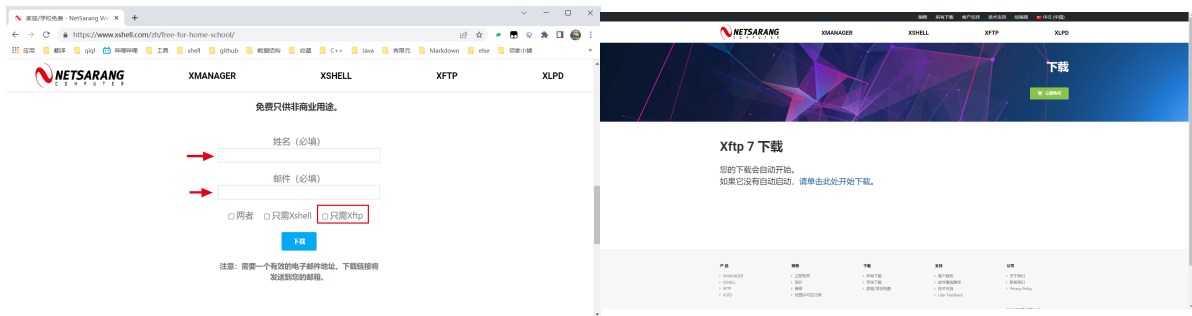
3.1.1 集群文件传输软件Xftp安装

Xftp 是一个功能强大的 SFTP、FTP 文件传输软件。使用 Xftp 可以方便的在 Windows 主机和远程 Linux 集群之间传文件

浏览器下载 Xftp

如果在刚才下载 Xshell 的页面中已经勾选了两者，那么邮箱中应该还有一个下载Xftp的链接，可以直接点击该链接进行下载，或者在浏览器输入<https://www.xshell.com/zh/free-for-home-school/>，跳转到如下界面，输入姓名及**有效**邮箱，勾选 Xftp，点击下载

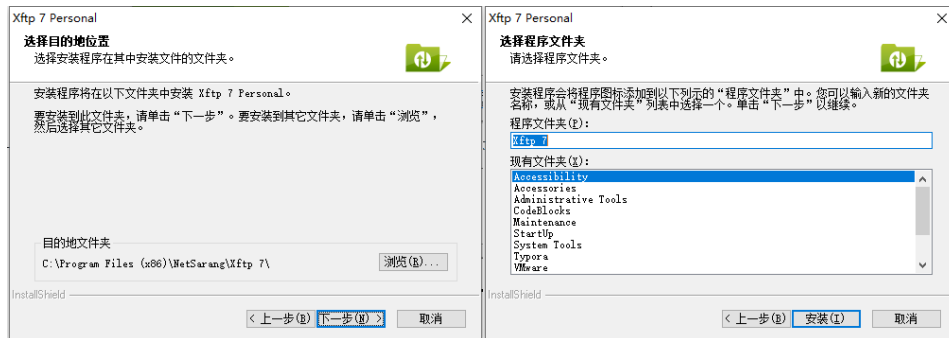
点击邮箱中的下载链接，进入以下界面，浏览器自动下载



打开 Xftp 7 安装包，点击下一步，勾选我接受，点击下一步



选择安装路径，也可以不更改安装路径，点击下一步，点击安装

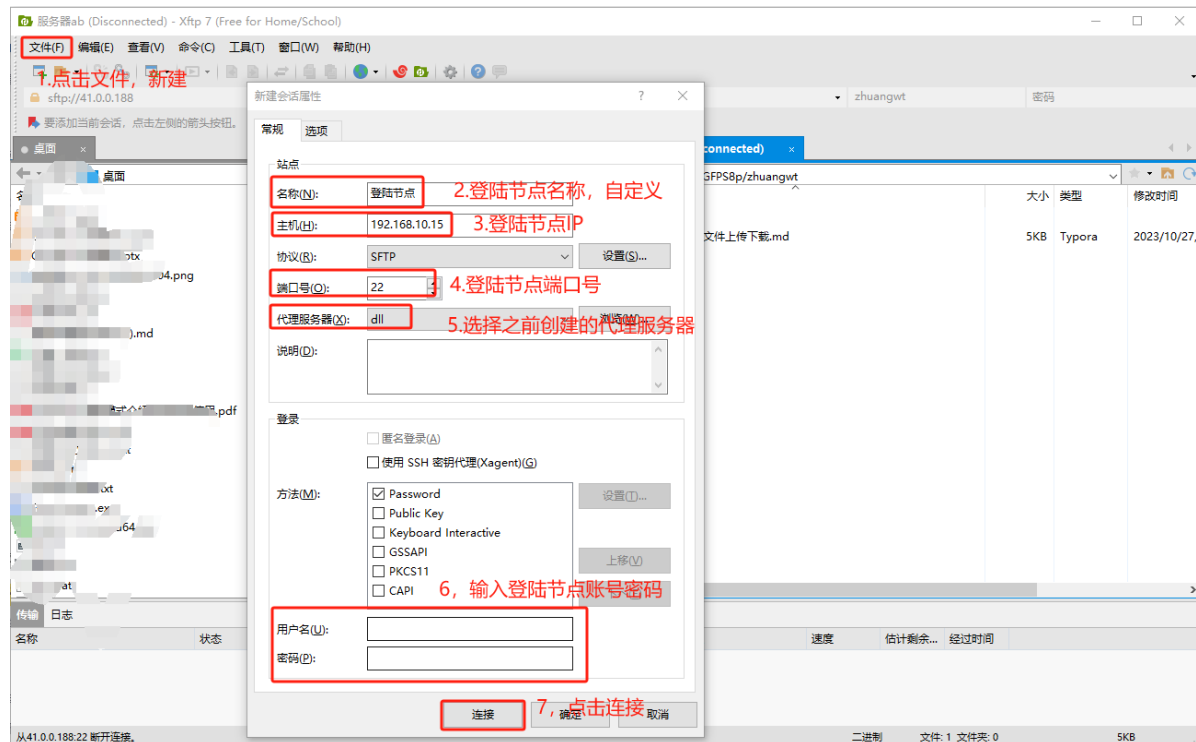


安装完成

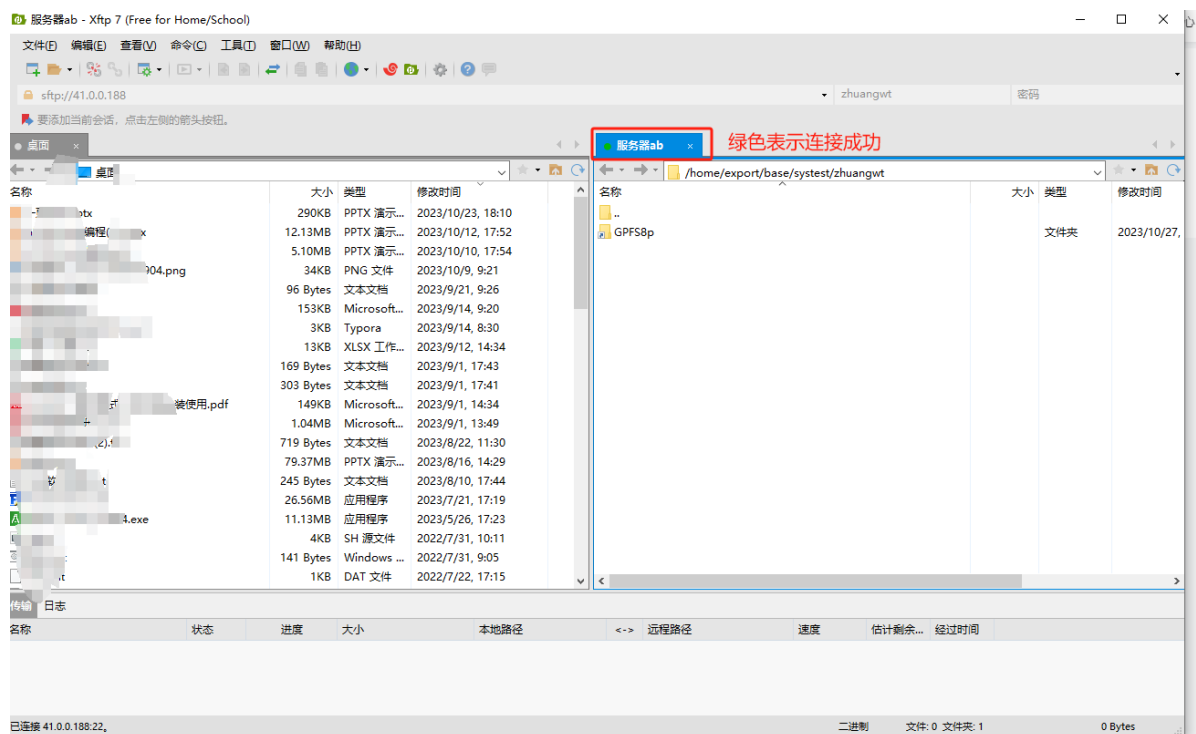


3.1.2 使用Xftp上传下载文件

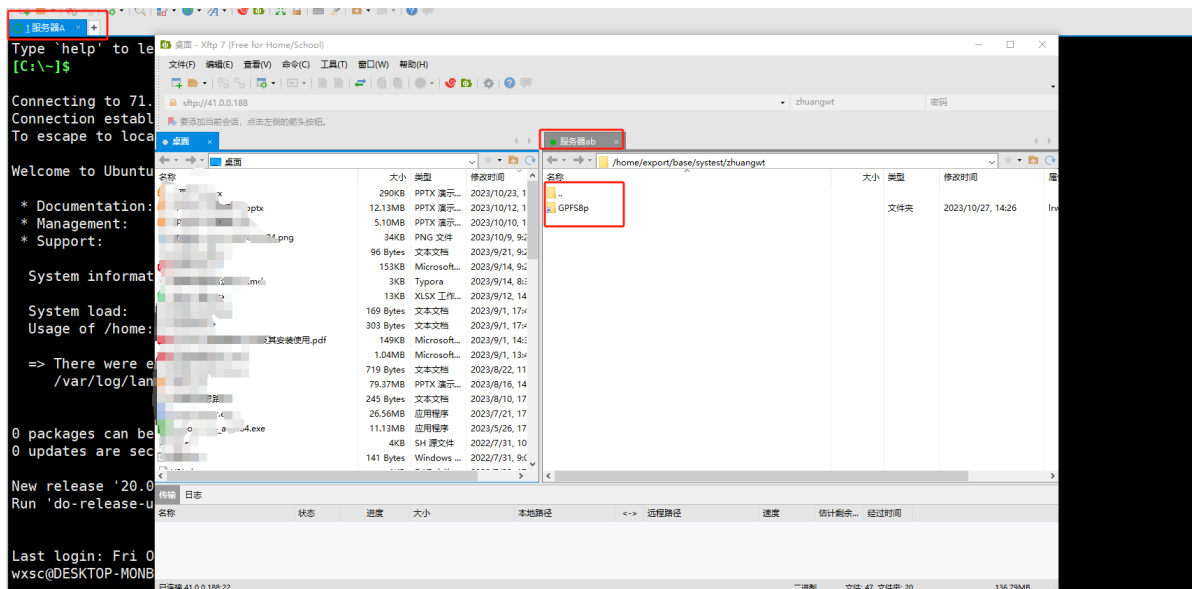
进入Xftp首页，点击左上角“文件”，点击“新建”，输入登陆节点的名称，IP：192.168.10.15，端口：22，名称可以任意，选择之前添加的代理服务器，输入登陆节点的“账号”和“密码”，点击“连接”，连接到登陆节点的文件系统



连接之后，绿色标识即表示连接成功



在之后的连接中，需要跳板机处于登陆状态下，在Xftp中选择登陆节点，可进入到登陆节点的文件系统中



3.2 Linux系统下的文件上传和下载

在linux系统下进行文件的上传和下载，我们可以一步实现文件的上传和下载，即采用 `scp` 的 `-o ProxyJump=` 选项，来指定跳板机的地址和用户名，利用跳板机向登陆节点上传或下载文件。

上传文件的命令格式为：

```
scp -o ProxyJump=username1@172.16.108.134 /root/data/file
username2@192.168.10.15:~
```

`/root/data/file` 为本地文件，此命令是将该文件 `file` 从本地上传到登陆节点的用户目录 `~` 下。

下载文件的命令格式为：

```
scp -o ProxyJump=username1@172.16.108.134 -r username2@192.168.10.15:~/file ~
```

`~/file` 是登陆节点用户目录下 `file` 文件的路径，此命令是将登陆节点下的 `file` 文件下载到本地用户目录 `~` 下。

`172.16.108.134` 为跳板机的主机地址，`username1` 为登陆跳板机时使用的用户名。

`192.168.10.15` 为登陆服务器的主机地址，`username2` 为登陆服务器时使用的用户名。

四、集群任务提交与管理

4.1任务提交

4.1.1 编写任务提交脚本

新建文件 `run.sh`，输入以下内容：

```
#!/bin/bash
#SBATCH -p x86_64_GPU
#SBATCH -n 1
#SBATCH -G 1
#SBATCH -o job.out

python a.py
```

- `#!/bin/bash` , 固定内容, 不用修改
- `#SBATCH -p x86_64_GPU` , 向调度系统申请 x86_64_GPU 队列的计算资源。调整队列名, 即更换队列运行计算任务。
- `#SBATCH -n 1` , 申请1个核
- `#SBATCH -G 1` , 申请1块GPU卡
- `#SBATCH -o job.out` , 程序的运行输出保存在 job.out 文件, 该文件文件名可以随意修改
- `python a.py` 程序自身的运行命令

4.1.2 提交计算任务

执行 `sbatch run.sh` 提交计算任务

```
[admin@manage tmp]$ sbatch run.sh
Submitted batch job 3656
```

4.1.3 查看任务状态

执行 `squeue -u 用户名` 查看计算任务运行状态

```
[admin@manage tmp]$ squeue -u admin
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES
NODELIST(REASON)						
3656	q_amd_sha	run.sh	admin	R	0:23	10 bn[054-063]

- JOBID, 任务编号
- ST, 任务状态, 作业状态包括 R (正在运行), PD (正在排队), CG (即将完成), CD (已完成)
- TIME, 运行时间
- NODES, 占用节点 (也称服务器, 下同) 个数
- NODELIST, 占用服务器的编号

4.1.4 查看程序输出

计算任务运行起来后, 执行 `tail -f job.out` 可以查看程序的实时输出, 执行 `Ctrl+C` 退出查看

```
[admin@manage tmp]$ tail -f job.out
Hello world from processor bn060, rank 896 out of 1280 processors
Hello world from processor bn060, rank 897 out of 1280 processors
```

4.2 任务管理

4.2.1 查看分区计算资源使用情况

```
[admin@manage tmp]$ sinfo
```

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
q_amd_share*	up	infinite	43	alloc	bn[023-050,334-336,345-356]
q_amd_share*	up	infinite	52	idle	bn[053-096,337-344]

节点状态包括:

`drain` (节点故障), `alloc` (节点在用), `idle` (节点可用), `down` (节点下线), `mix` (节点部分占用, 但仍有剩余资源)

4.2.2 计算任务停止

执行 `scancel 任务编号` 停止计算任务，“任务编号”通过 `squeue -u 用户名` 可以查到。

```
scancel 3656
```

4.2.3 计算任务详细信息查看

执行 `scontrol show job 任务编号` 可以查到计算任务的工作目录、输出文件等详细信息

```
scontrol show job 3644
```

4.3 python程序运行教程

4.3.1 进入目录，上传文件

登陆到登陆节点后，进入到data目录：

```
cd ~/data
```

上传python文件：

使用 xftp 软件上传您需要运行的python文件到登陆节点

4.3.2 编写作业提交脚本

在python程序同级目录下，编写作业提交脚本：

```
vim run.sh
```

进入vim后，点击键盘上的字母 i 进入编辑模式，输入以下内容

```
#!/bin/bash
#SBATCH -p x86_64_GPU
#SBATCH -n 1
#SBATCH -o %J.out
#SBATCH -G 1

module load Anaconda/mini3-23.1.0
module load cuda/12.1.0
source activate pytorch2.1.0_cuda12.1.0

python -u BipedalWalker2-sac.py
```

其中：

-p x86_64_GPU 表示将作业提交到 x86_64_GPU 队列

-n 1 表示申请一个核

-o %J.out 表示将作业的输出重定向到当前目录下的 %J.out 文件中，其中 %J 为作业提交之后的作业号，如作业号为1678，则输出文件即为 1678.out

-G 1 为作业申请一个 GPU

module load Anaconda/mini3-23.1.0 表示加载 miniconda3 模块

module load cuda/12.1.0 表示加载 cuda_12.1.0 模块

source activate pytorch2.1.0_cuda12.1.0 表示激活 conda 的虚拟环境，虚拟环境中的 pytorch 为2.1.0版本， cuda 为12.1.0版本，为作业运行做准备

python -u BipedalWalker2-sac.py 表示运行python程序，并将程序输出强制打印(不影响程序运行)

作业提交脚本编写完成后，操作保存并退出

4.3.3

作业提交脚本编写完成，使用如下进行作业提交操作

```
sbatch run.sh
```

作业提交后，终端会输出此次作业的作业号，
如：

```
Submitted batch job 2923
```

其中 2923 即为本次作业的作业号。

我们可以使用以下命令**查看作业运行情况**

```
tail -f 2923.out
```

五、存储使用情况查看

```
lfs quota -u zizhanghao /share -h
```

这个命令是查看用户自己的子账号的存储使用情况（**zizhanghao** 为自己的子账号用户名），用户可以自己执行的

```
lfs quota -g zuzhanghu /share -h
```

这个命令是查看组账户(用户组)的存储使用情况（**zuzhanghu** 为当前组账户名），用户也可以自己执行查看

六、任务账单查看

计费系统网址：<https://172.16.108.134:8090/>

登录后可查询相应的作业和账单

账号：**集群子账号名 / 组账户名**

初始密码：**para@1234**

注意：登录后请及时修改密码

用户账单: dongrunmin

用户账单

消息中心

2025年01月

2024年12月

2024年11月

2024年10月

2024年09月

2024年08月

自定义时间

未计费

导出

自定义结束时间: 2024-01-10 - 2025-01-16 周 月 全年

所属账户: -- 作业数量: 0 CPU核时数: 0.000 GPU卡时数: 0.000 作业总费用(元): 0.00 计费周期: 2024-01-10~2025-01-16

作业ID: 请输入作业ID

用户名称: 请输入用户名称

集群: 全部

集群账号: 请输入集群账号

队列名称: 全部

作业状态: COMPLETED

计费模式: 全部

计费对象: 全部

隐藏更多选项

重置

作业ID	用户名称	集群	集群账号	提交时间	开始时间	结束时间	运行时长	作业状态	队列	计费模式	计费对象	CPU核数	CPU核时数	GPU卡数
30057		cluster		2024-06-11 21:13:44	2024-06-11 21:13:46	2024-06-11 21:13:58	12.00s	COMPLETED	gpu_ai	计时	GPU(卡时)	16	0.053	1
30056		cluster		2024-06-11 21:12:57	2024-06-11 21:12:58	2024-06-11 21:13:18	20.00s	COMPLETED	gpu_ai	计时	GPU(卡时)	16	0.089	1
27565		cluster		2024-05-16 18:54:18	2024-05-17 21:16:06	2024-05-19 10:29:45	1d13h13m39s	COMPLETED	gpu_ai	计时	GPU(卡时)	4	148.910	1
27564		cluster		2024-05-16 18:54:18	2024-05-17 21:19:06	2024-05-19 10:31:34	1d13h22m28s	COMPLETED	gpu_ai	计时	GPU(卡时)	4	149.498	1