



上海科技大学  
ShanghaiTech University

# CS272-Computer Vision II Tutorial: Introduction to AI Cluster

常琬星



立志成才报国裕民

# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
- 安装PyTorch
- 提交作业
- 远程断点调试

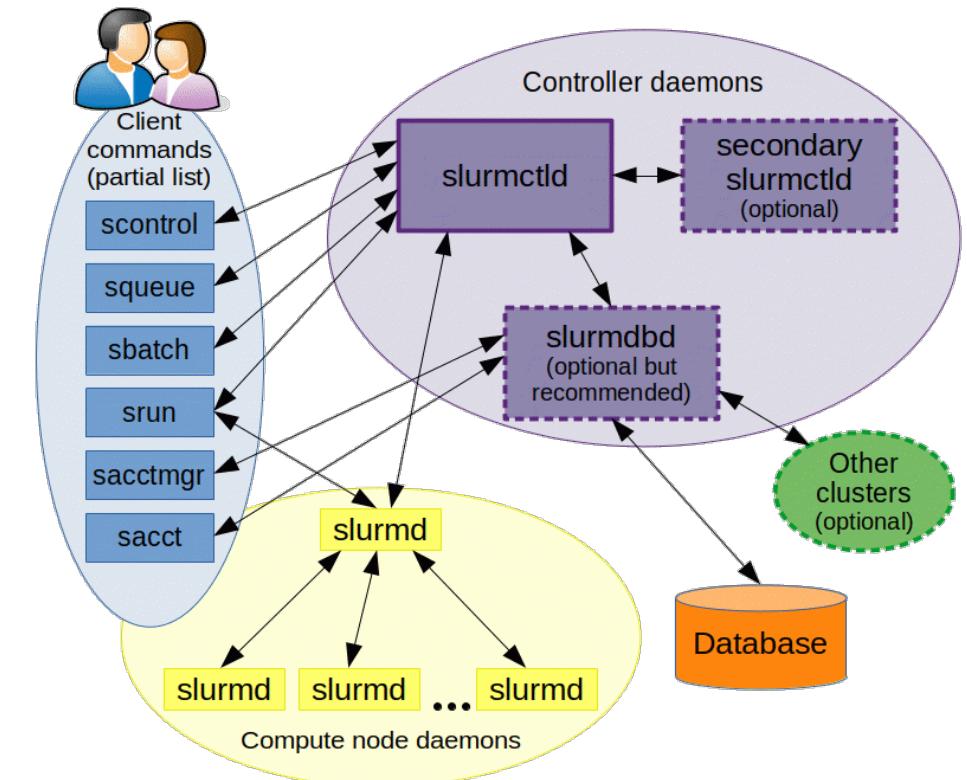


立志成才报国裕民

# AI Cluster介绍 : Slurm调度系统



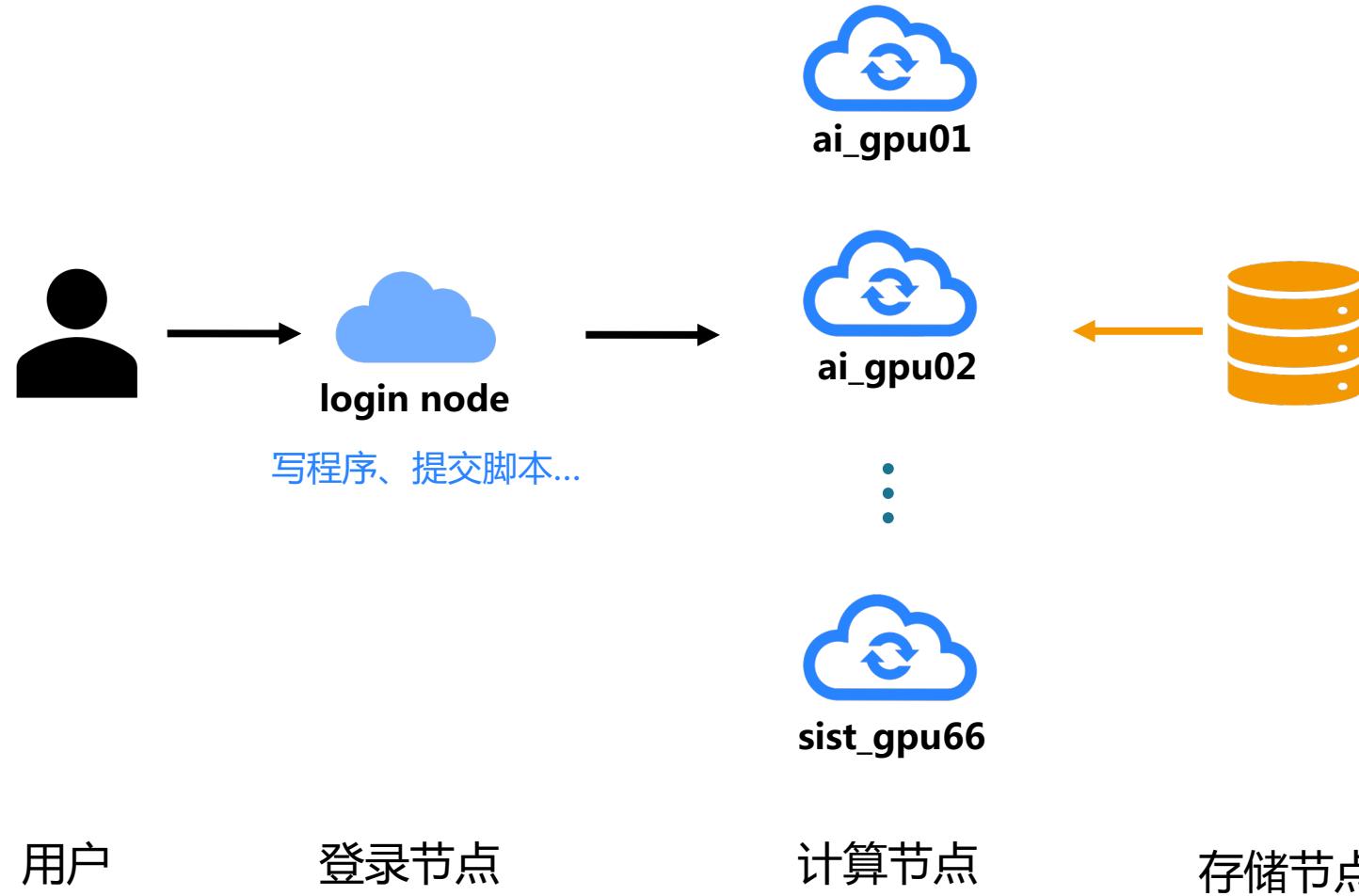
- 集群基于Slurm调度系统
  - Slurm (**S**imple **L**inux **U**nity for **R**esource **M<http://slurm.schedmd.com/>)**
  - Linux集群超级计算系统资源管理和作业调度系统
- ✓ **每节点独立运行**，仅节点上运行的任务会产生IO开销，并且不直接依赖于存储，系统的稳定性更强。
- ✓ 用户通过系统账号登录，配合NIS可以方便实现基于用户和用户组的目录权限管理。
- ✓ Slurm本身支持**任务的资源调度和排队**系统，能更好地分配资源。
- ✓ Slurm支持历史作业信息导出，方便收集和统计数据。



# AI Cluster介绍：用户视角



上海科技大学  
ShanghaiTech University



立志成才报国裕民

# 注意事项



上海科技大学  
ShanghaiTech University

- 信息学院AI Cluster用户手册：<http://10.15.89.177:8889/index.html>
- 必须连校园网才能访问，校外同学请使用上科大VPN：  
<https://library.shanghaitech.edu.cn/4032/list.htm>
- 禁止占用资源却不运行**，会被直接取消任务，情节严重的会对用户发送警告并暂停权限。
  - 请注意，使用salloc命令占用节点资源，请确保资源分配后会及时运行作业。
- 禁止在登录节点运行计算程序**
- 账号仅作为CS272课程作业和课程project使用，**禁止另作他用，违者必究！**
- 集群账号仅供申请主体的申请人本人使用，不得出借账号
- 请妥善保管好账号密码或密钥以免泄露。
- 请仅通过两个登录节点登录到集群。



立志成才报国裕民



实时GPU使用情况监控页面：<http://10.15.89.177:8899/gpu>

ai\_gpu01 2022-09-27 15:38:02 (CPU:12.17% [10/80] MEM:66G/376G)

#	Name	Memory	GPU	Processes	pid@user (RAM)
0	NVIDIA A40	<div style="width: 75%;">75%</div>	<div style="width: 100%;">100%</div>	218972@ shicheng	(75%)
1	NVIDIA A40	<div style="width: 57%;">57%</div>	<div style="width: 100%;">100%</div>	218973@ shicheng	(57%)
2	NVIDIA A40	<div style="width: 91%;">91%</div>	<div style="width: 99%;">99%</div>	218975@ shicheng	(91%)
3	NVIDIA A40	<div style="width: 68%;">68%</div>	<div style="width: 99%;">99%</div>	218976@ shicheng	(68%)
4	NVIDIA A40	<div style="width: 90%;">90%</div>	<div style="width: 98%;">98%</div>	218977@ shicheng	(90%)
5	NVIDIA A40	<div style="width: 75%;">75%</div>	<div style="width: 99%;">99%</div>	218978@ shicheng	(75%)
6	NVIDIA A40	<div style="width: 92%;">92%</div>	<div style="width: 98%;">98%</div>	218979@ shicheng	(92%)
7	NVIDIA A40	<div style="width: 68%;">68%</div>	<div style="width: 100%;">100%</div>	218980@ shicheng	(68%)

ai\_gpu02 2022-09-27 15:38:02 (CPU:13.14% [11/80] MEM:92G/376G)

#	Name	Memory	GPU	Processes	pid@user (RAM)
0	NVIDIA A40	<div style="width: 47%;">47%</div>	<div style="width: 0%;">0%</div>	25130@ huanghz 27900@ huanghz	(28%) (2%) (2%) (2%)
				26772@ huanghz 28057@ huanghz	(2%) (2%)
1	NVIDIA A40	<div style="width: 21%;">21%</div>	<div style="width: 100%;">100%</div>	26772@ huanghz	(21%)
2	NVIDIA A40	<div style="width: 27%;">27%</div>	<div style="width: 0%;">0%</div>	26852@ huanghz	(27%)
3	NVIDIA A40	<div style="width: 14%;">14%</div>	<div style="width: 0%;">0%</div>	27288@ huanghz	(14%)
4	NVIDIA A40	<div style="width: 29%;">29%</div>	<div style="width: 0%;">0%</div>	27900@ huanghz	(29%)
5	NVIDIA A40	<div style="width: 19%;">19%</div>	<div style="width: 100%;">100%</div>	28057@ huanghz	(19%)
6	NVIDIA A40	<div style="width: 38%;">38%</div>	<div style="width: 100%;">100%</div>	28110@ huanghz	(38%)
7	NVIDIA A40	<div style="width: 24%;">24%</div>	<div style="width: 0%;">0%</div>	28139@ huanghz	(24%)



立志成才报国裕民

# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
- 安装PyTorch
- 提交作业
- 远程断点调试



立志成才报国裕民

# 登录集群



上海科技大学  
ShanghaiTech University

- 已为之前申请集群账号的同学已经开通权限，用户名为邮箱前缀+“-cs272”，默认密码为sist，登陆后请使用yppasswd修改默认密码。
  - 如邮箱 [changwx@shanghaitech.edu.cn](mailto:changwx@shanghaitech.edu.cn) 对应的集群账号用户名为**changwx-cs272**
- 账号都开通了CS272队列权限，队列限制同时使用2张显卡，任务时长最大2天。
- 集群使用请参考<http://10.15.89.177:8889/>



立志成才报国裕民

# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
  - 终端工具登录
  - 开发工具登录
- 安装PyTorch
- 提交作业
- 远程断点调试



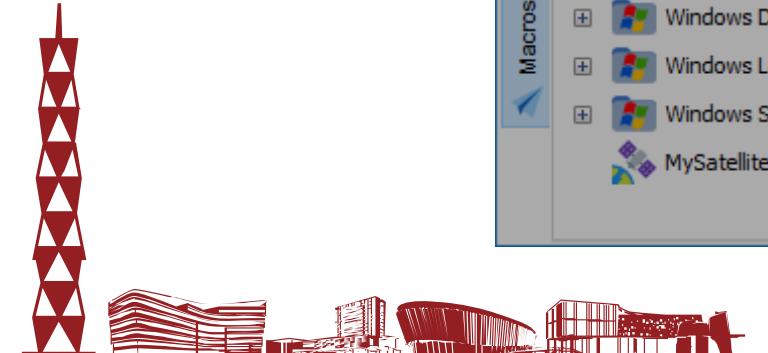
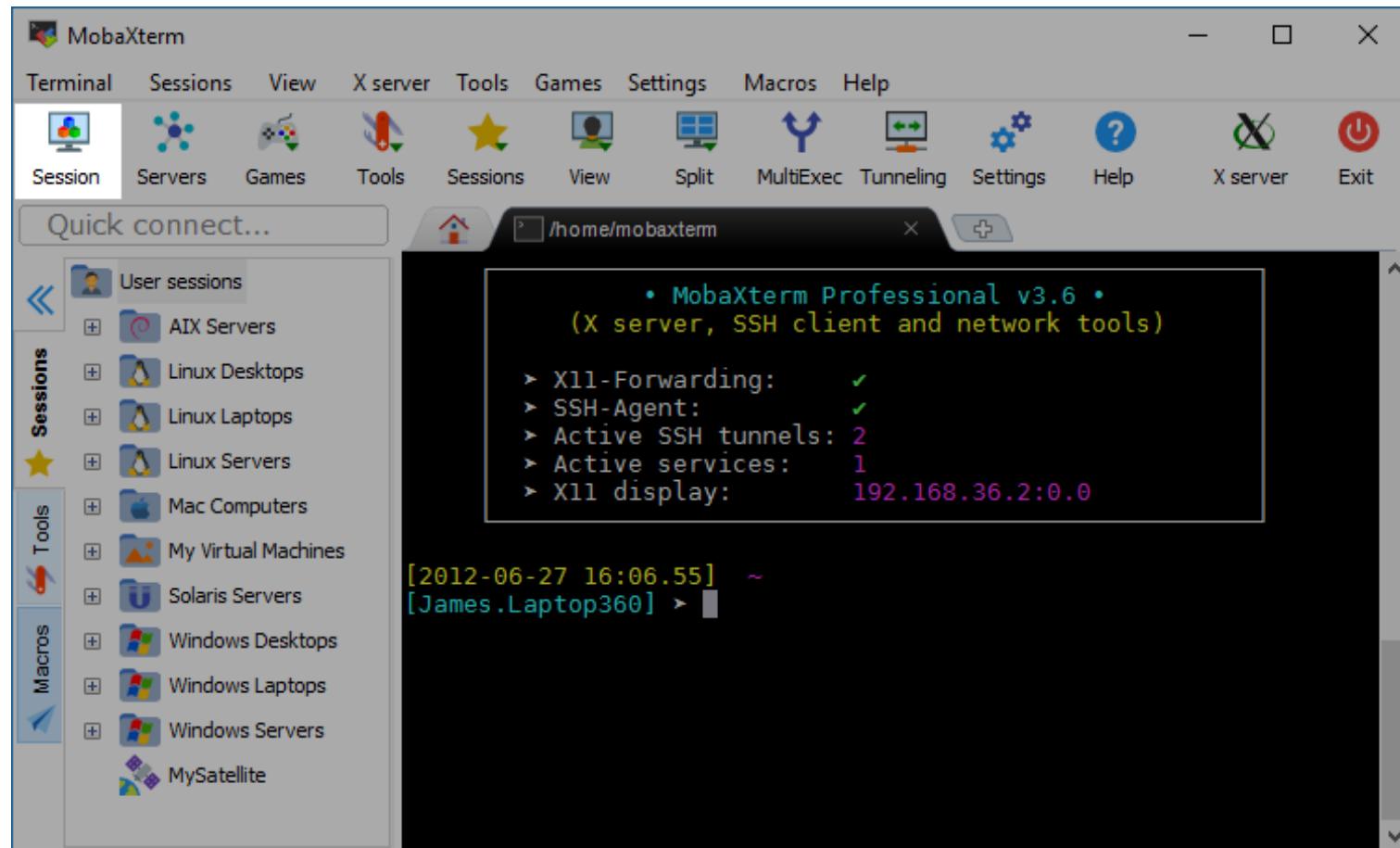
立志成才报国裕民

# Windows: MobaXterm 远程登录



上海科技大学  
ShanghaiTech University

(a) 启动 mobaxterm，并在菜单栏的左上角找到 session

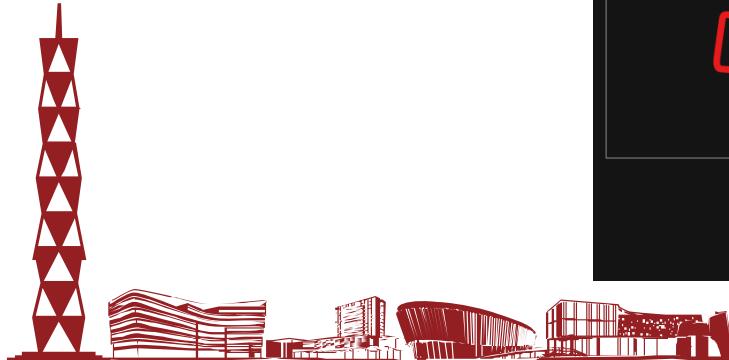
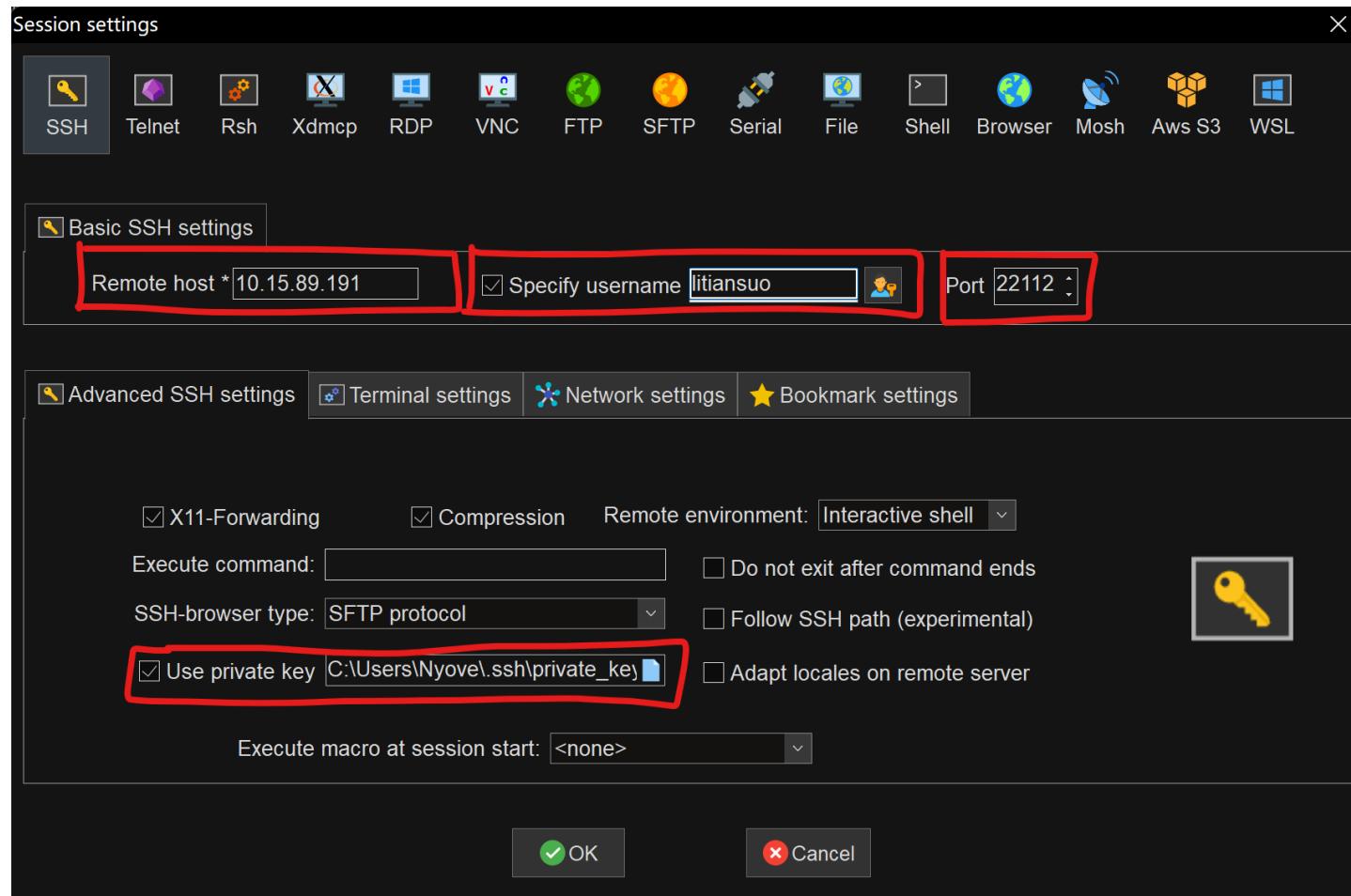


立志成才报国裕民

# Windows: MobaXterm 远程登录



(b) 之后依次填入 远程ip 用户名 端口 私钥文件路径



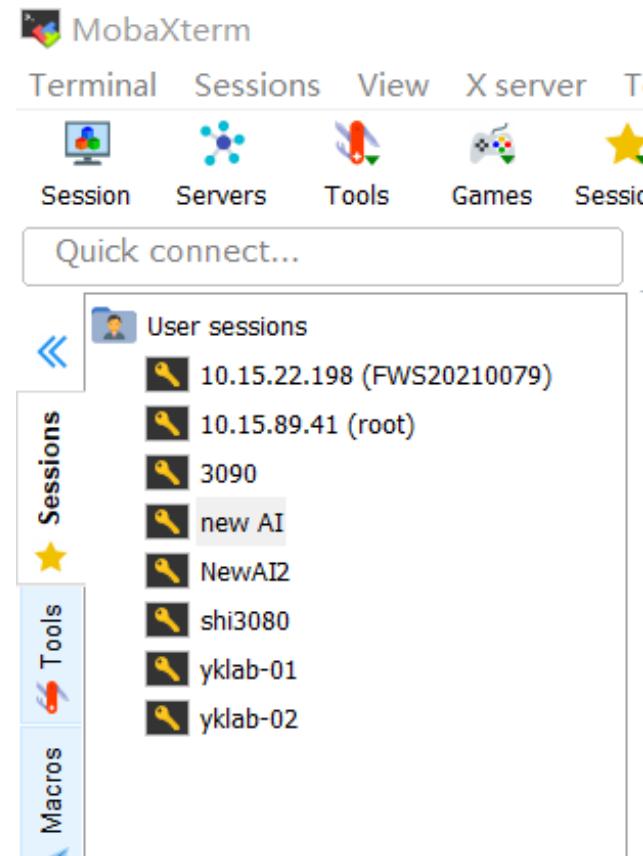
立志成才报国裕民

# Windows: MobaXterm 远程登录



上海科技大学  
ShanghaiTech University

(c) 最后双击已添加的 session 登录。



立志成才报国裕民



用户可以使用终端中的命令行工具登录。下列语句指出了该节点的IP地址、用户名和SSH端口。

```
$ ssh YOUR_USERNAME@TARGET_IP:PORT -i <Path of private key>
```

如：

```
$ ssh litiansuo@10.15.89.191:22112 -i ~/.ssh/shanghaitech_hpc_key"
```



注意：出于安全性的考虑， ssh 需要相对低的文件权限，请使用命令 `chmod 500 <file>`。有些情况下，当主机名重复的时候 ssh 连接会出现错误，请删除 `known_hosts` 中对应的行。通常情况下，使用 `rm ~/.ssh/known_hosts` 可以解决为了方便，长期使用更推荐使用 `ssh-config-file`，查看 [这个网页](https://linuxize.com/post/using-the-ssh-config-file/) 以获得更多信息。

<https://linuxize.com/post/using-the-ssh-config-file/>



立志成才报国裕民

# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
  - 终端工具登录
  - 开发工具登录
- 安装PyTorch
- 提交作业
- 远程断点调试



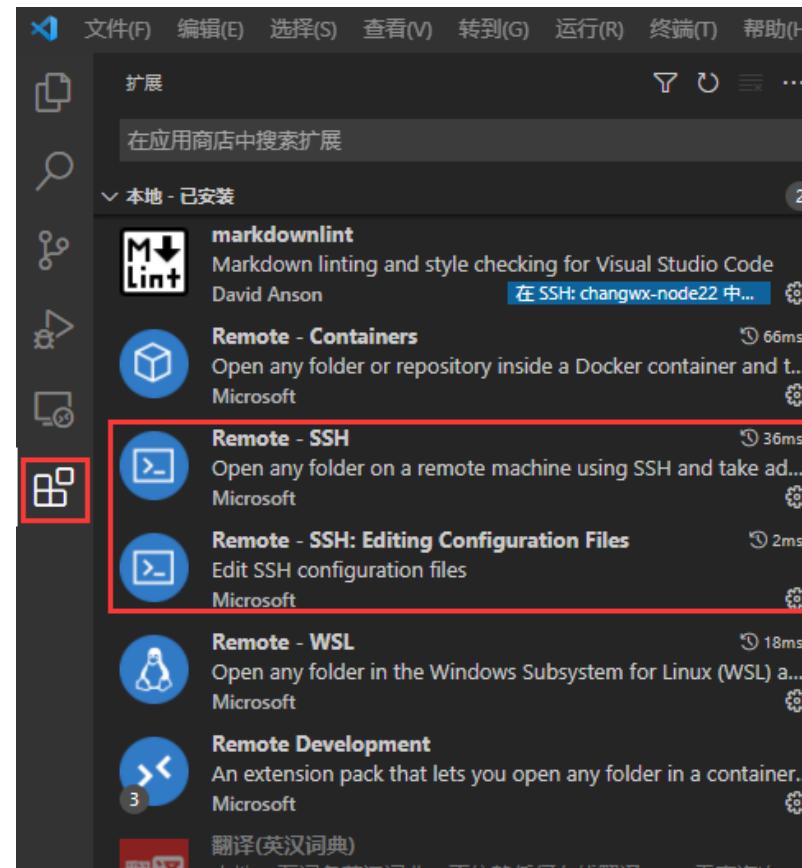
立志成才报国裕民

# VSCODE 远程登录



上海科技大学  
ShanghaiTech University

(a) 安装拓展



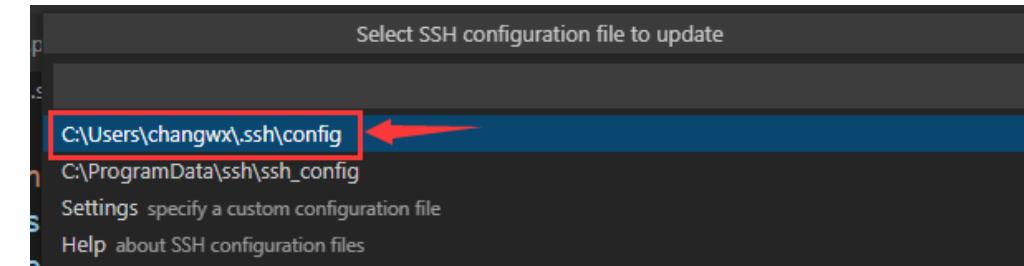
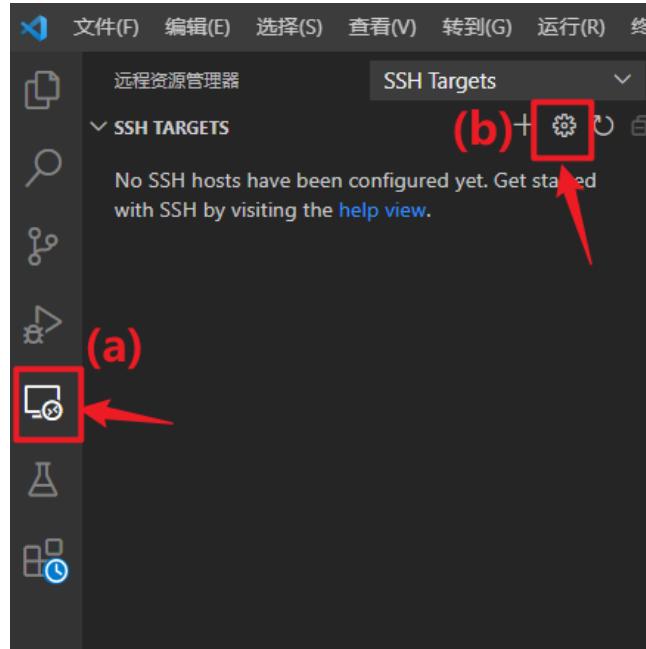
立志成才报国裕民

# VSCODE 远程登录



上海科技大学  
ShanghaiTech University

## (b) 配置节点



```
C: > Users > changwx > .ssh > config
65
66 # new AI
67 Host changwx-login01 自定义名字
68     HostName 10.15.89.191 login01节点ip
69     User changwx 集群username
70     Port 22112 端口
71
```

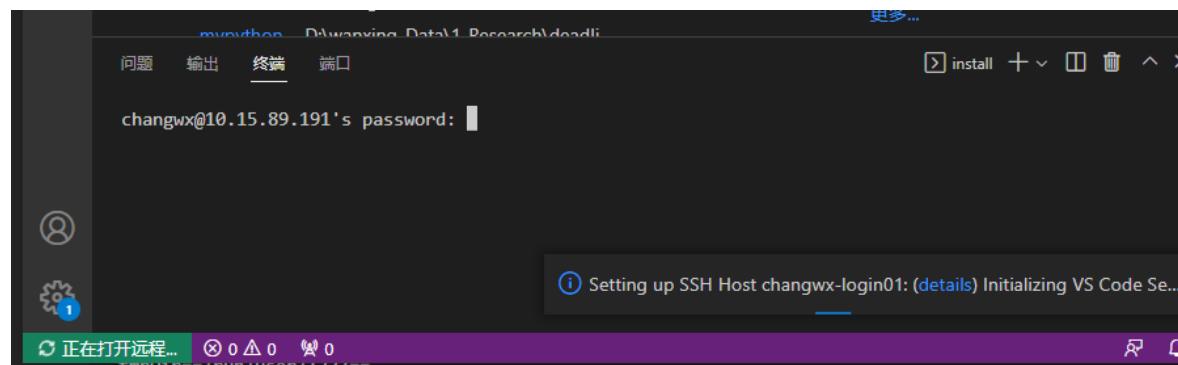
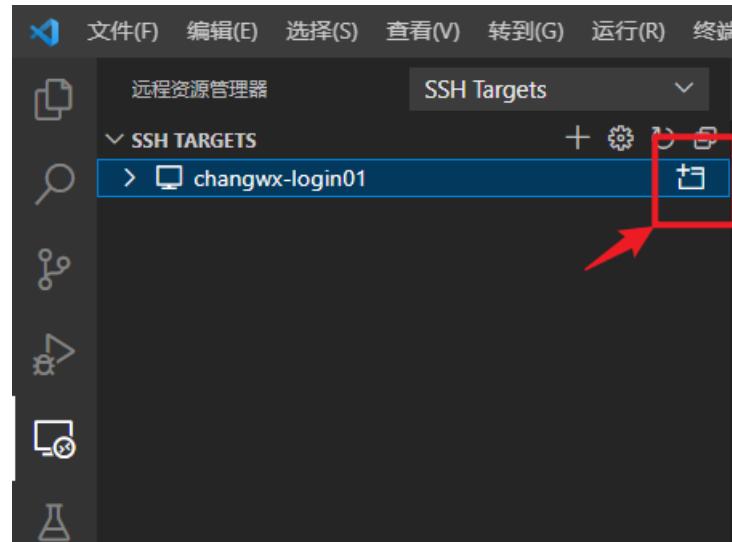
立志成才报国裕民

# VSCODE 远程登录



上海科技大学  
ShanghaiTech University

(c) 配置完成，登录节点，输入密码即可



立志成才报国裕民

# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
- 安装PyTorch
- 提交作业
- 远程断点调试



立志成才报国裕民

# 安装PyTorch



## (a) 在~/.bashrc export环境

在"/public/software/bashrc\_example"可以复制到一些例子

```
.bashrc
1 # .bashrc
2
3 # Source global definitions
4 if [ -f /etc/bashrc ]; then
5     . /etc/bashrc
6 fi
7
8 # Uncomment the following line if you don't like systemctl's auto-paging feature:
9 # export SYSTEMD_PAGER=
10
11 # User specific aliases and functions
12 #export PATH="$HOME/bin:$PATH"
13 #zsh
14
15 #####ANACONDA3#####
16 export ANACONDA3_HOME=/public/software/anaconda3
17 export PATH=$PATH:$ANACONDA3_HOME/bin
18
19 #####CUDA 11.4#####
20 export PATH=/public/software/CUDA/cuda-11.4/bin${PATH:+:$PATH}
21 export LD_LIBRARY_PATH=/public/software/CUDA/cuda-11.4/lib64${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
22
23 #####GCC 10.2#####
24 export PATH=/public/software/gcc/gcc-10.2/bin:$PATH
25 export LD_LIBRARY_PATH=/public/software/mpc/mpc=1.2.1/lib:/public/software/gmp/gmp-6.2.1/lib:/public/
   /software/mpfr/mpfr-4.1.0/lib:$PATH
26
```

\*注意，每次改完~/.bashrc或者~/.bash\_profile或者~/.zshrc这类文件，最好重启命令行看看是不是成功修改了，不重启的话可能不会生效。或者用source命令重新处理一下这些文件。



# 安装PyTorch



(b) 输入命令`conda init bash`

在`~/.bashrc`里面会自动生成以下行

```
1 # .bashrc
2
3 # Source global definitions
4 if [ -f /etc/bashrc ]; then
5   . /etc/bashrc
6 fi
7
8 # Uncomment the following line if you don't like systemctl's auto-paging feature:
9 # export SYSTEMD_PAGER=
10
11 # User specific aliases and functions
12 #export PATH="$HOME/bin:$PATH"
13 #zsh
14
15 #####ANACONDA3#####
16 export ANACONDA3_HOME=/public/software/anaconda3
17 export PATH=$PATH:$ANACONDA3_HOME/bin
18
19 #####CUDA 11.4#####
20 export PATH=/public/software/CUDA/cuda-11.4/bin${PATH:+:$PATH}
21 export LD_LIBRARY_PATH=/public/software/CUDA/cuda-11.4/lib64${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
22
23 #####GCC 10.2#####
24 export PATH=/public/software/gcc/gcc-10.2/bin:$PATH
25 export LD_LIBRARY_PATH=/public/software/mpc/mpc=1.2.1/lib:/public/software/gmp/gmp-6.2.1/lib:/public/software/mpfr/mpfr-4.1.0/lib:$PATH
26
27
28 # >>> conda initialize >>
29 # !!! Contents within this block are managed by 'conda init' !!
30 __conda_setup="$('/public/software/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
31 if [ $? -eq 0 ]; then
32   eval "$__conda_setup"
33 else
34   if [ -f "/public/software/anaconda3/etc/profile.d/conda.sh" ]; then
35     . "/public/software/anaconda3/etc/profile.d/conda.sh"
36   else
37     export PATH="/public/software/anaconda3/bin:$PATH"
38   fi
39 fi
40 unset __conda_setup
41 # <<< conda initialize <<<
42
```



立志成才报国裕民

# 安装PyTorch



(c) 输入命令 `conda init zsh`

```
.zshrc
79 # Add wisely, as too many plugins slow down shell startup.
80 plugins=(git)
81
82 source $ZSH/oh-my-zsh.sh
83
84 # User configuration
85
86 # export MANPATH="/usr/local/man:$MANPATH"
87
88 # You may need to manually set your language environment
89 # export LANG=en_US.UTF-8
90
91 # Preferred editor for local and remote sessions
92 # if [[ -n $SSH_CONNECTION ]]; then
93 #     export EDITOR='vim'
94 # else
95 #     export EDITOR='mvim'
96 # fi
97
98 # Compilation flags
99 # export ARCHFLAGS="-arch x86_64"
100
101 # Set personal aliases, overriding those provided by oh-my-zsh libs,
102 # plugins, and themes. Aliases can be placed here, though oh-my-zsh
103 # users are encouraged to define aliases within the ZSH_CUSTOM folder.
104 # For a full list of active aliases, run `alias`.
105 #
106 # Example aliases
107 # alias zshconfig="mate ~/.zshrc"
108 # alias ohmyzsh="mate ~/.oh-my-zsh"
109
110 # To customize prompt, run `p10k configure` or edit ~/.p10k.zsh.
111 [[ ! -f ~/.p10k.zsh ]] || source ~/.p10k.zsh
112
113 # >>> conda initialize >>>
114 # !! Contents within this block are managed by 'conda init' !!
115 __conda_setup="$('/public/software/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev/null)"
116 if [ $? -eq 0 ]; then
117     eval "$__conda_setup"
118 else
119     if [ -f "/public/software/anaconda3/etc/profile.d/conda.sh" ]; then
120         . "/public/software/anaconda3/etc/profile.d/conda.sh"
121     else
122         export PATH="/public/software/anaconda3/bin:$PATH"
123     fi
124 fi
125 unset __conda_setup
126 # <<< conda initialize <<<
127
```



# 安装PyTorch



上海科技大学  
ShanghaiTech University

(d) 重启命令行

\*例子以username为changwx展示

(e) 尝试自带的anaconda的环境

```
Last login: Tue Jan 18 16:44:39 2022 from 10.19.126.210
(base) [changwx@login01 ~]$ which python
/public/software/anaconda3/bin/python
(base) [changwx@login01 ~]$ python --version
Python 3.8.8
(base) [changwx@login01 ~]$ █
```

(f) 发现没有安装pytorch

```
Python 3.8.8
(base) [changwx@login01 ~]$ python
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import torch
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'torch'
>>> █
```

(g) 安装自己的子环境

```
conda create --name changwx python=3.7      # python=3.7 指定python版本
```

(h) 输入命令 conda activate changwx

在.bashrc和.zshrc都写一下 “conda activate changwx”

```
(base) [changwx@login01 ~]$ zsh
> conda activate changwx
> which python
/public/home/changwx/.conda/envs/changwx/bin/python
> python --version
Python 3.7.12
```



立志成才报国裕民

# 安装PyTorch



(i) 装pytorch之前配置清华源

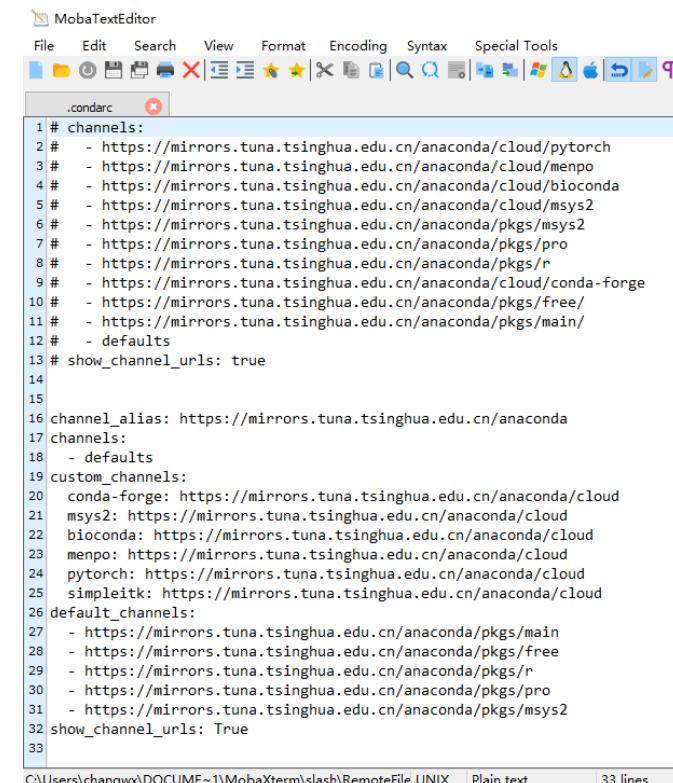
\*例子以username为changwx展示

刚开始是没有"/public/home/changwx/.condarc"这个文件的。要先运行一句

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge
```

以后就自动生成`~/.condarc`这个文件了。然后打开这个文件把复制下面这段全部复制进来

```
channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda
channels:
- defaults
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
default_channels:
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
show_channel_urls: True
```



# 立志成才报国裕民

# 安装PyTorch



(j) 开始安装pytorch cuda=11.4，输入以下命令：

```
conda install pytorch torchvision cudatoolkit=11.4
```

【保证有conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge】

done就是安装成功了

```
liblapack-3.9.0      | 12 KB      | #####| 100%
sleef-3.5.1          | 1.5 MB     | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: \ By downloading and using the CUDA Toolkit conda packages, you accept the terms and conditions of the CUDA End User License Agreement (EULA): https://docs.nvidia.com/cuda/eula/index.html
By downloading and using the cuDNN conda packages, you accept the terms and conditions of the NVIDIA cuDNN EULA - 
https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
do
```

(k) 检查一下是否安装成功，输入以下命令

```
python
import torch
print(torch.__version__)
```

```
> python
Python 3.7.12 | packaged by conda-forge | (default, Oct 26 2021, 06:08:21)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> print(torch.__version__)
1.10.0
```



# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
- 安装PyTorch
- 提交作业
- 远程断点调试



立志成才报国裕民

# 常用命令



上海科技大学  
ShanghaiTech University

- 请查阅集群手册：<http://10.15.89.177:8889/job/index.html>
- 请重点查阅以下几个常用命令：
  - sbatch
  - salloc
  - srun
  - squeue
  - scancel



立志成才报国裕民

# 提交作业脚本（单卡）



上海科技大学  
ShanghaiTech University

- 这是一个名为run.sh的作业脚本，该脚本向debug队列申请1个节点4核，并在作业运行时开始邮件通知。
- 在命令行提交作业：sbatch run.sh

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p debug
#SBATCH --cpus-per-task=4
#SBATCH --mail-type=all
#SBATCH --mail-user=YOU@MAIL.COM
#SBATCH -N 1
#SBATCH -t 3:00
#SBATCH --gres=gpu:1
#SBATCH --output=%j.out
#SBATCH --error=%j.err

dataset='domainnet'
source='painting'
target='sketch'

python main.py --dataset ${dataset} --source ${source} --target ${target}
```



立志成才报国裕民

# 提交作业脚本（多卡）

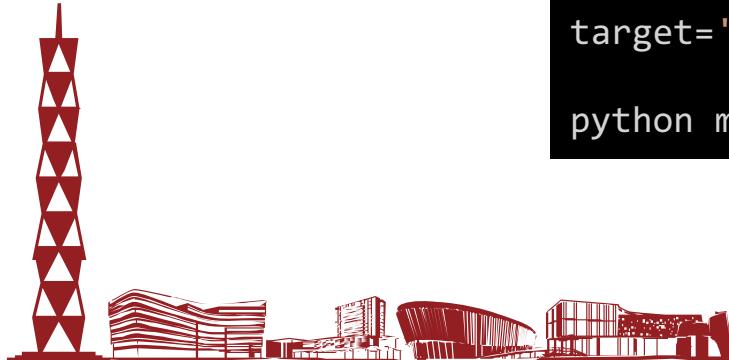


- 以下作业请求2张GPU卡，其中1个CPU进程管理1张GPU卡。

```
#!/bin/bash
#SBATCH --job-name=GPU
#SBATCH --partition=debug
#SBATCH -n 2
#SBATCH --ntasks-per-node=2
#SBATCH --gres=gpu:2
#SBATCH --mail-type=end
#SBATCH --mail-user=YOU@MAIL.COM
#SBATCH --output=%j.out
#SBATCH --error=%j.err

dataset='domainnet'
source='painting'
target='sketch'

python main.py --dataset ${dataset} --source ${source} --target ${target}
```



# Outline



上海科技大学  
ShanghaiTech University

- AI Cluster介绍
- 登录集群
- 安装PyTorch
- 提交作业
- 远程断点调试



立志成才报国裕民

# VSCODE slurm 断点调试



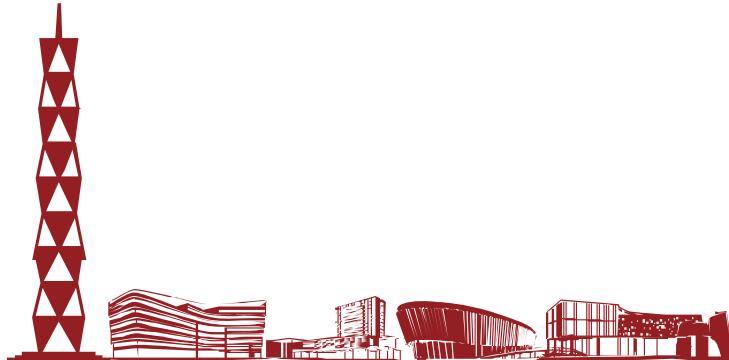
\*例子以username为changwx展示

(a) 在登录节点申请debug队列的一张卡

命令 : `salloc -N 1 -n 4 -p debug --gres=gpu:1`

```
> salloc -N 1 -n 4 -p debug --gres=gpu:1
salloc: Pending job allocation 15625
salloc: job 15625 queued and waiting for resources
salloc: job 15625 has been allocated resources
salloc: Granted job allocation 15625
(changwx) [changwx@login01 ~]$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
15625    debug   bash changwx R   0:03      1 sist_gpu53
15120  normal v6_mu09_ changwx R  0:57:42      1 sist_gpu64
15127  normal v6_mu09_ changwx R 14:36:07      1 sist_gpu63
15126  normal v6_mu08_ changwx R 21:18:07      1 sist_gpu63
15125  normal v6_mu08_ changwx R 23:11:46      1 sist_gpu65
15121  normal v6_mu06_ changwx R 23:56:34      1 sist_gpu64
(changwx) [changwx@login01 ~]$ ^C
```

这个sist\_gpu53要写进ssh文件



# VSCODE slurm 断点调试



(b) 配置ssh文件【每次申请新节点，都要改计算节点的HostName】

\*例子以username为changwx展示

The screenshot shows the Visual Studio Code interface. On the left, the 'SSH Targets' sidebar is open, displaying a list of nodes under 'SSH TARGETS'. A specific node, 'changwx-c02n01', is highlighted with a red box and labeled '(a)'. Another node, 'changwx-admin', has its 'SSH Targets' expanded, showing sub-nodes like 'p300' and 'DANCE /p300/code/UniversalDA'. A third node, 'changwx-login01', also has its 'SSH Targets' expanded, showing 'code /public/home/changwx/p300'. On the right, the main editor area shows the 'launch.json' file for the 'SSH: changwx-login01' configuration. A red box highlights the path 'C:\Users\changwx\ssh\config' in the 'Select SSH configuration file to update' dropdown menu, labeled '(b)'. Another red box highlights the file path 'C:\Users\changwx\ssh\config' in the code editor, labeled '(c)'. The code in the editor is as follows:

```
22 // " :> C:\Users\changwx\ssh\config
23 // C:\ProgramData\ssh\sshd_config
24 // Settings specify a custom configuration file
25 // Help about SSH configuration files
26 // "--target", "amazon",
27 // "--dataset", "officehome",
28 // "--topk", "3",
29 // "--gpu", "0"
30 // ],
31 {
32     "name": "launch",
33     "type": "python",
34 }
```

The screenshot shows a terminal or code editor window displaying an 'ssh config' file. It contains three host definitions:

- Host 'changwx-login01':  
HostName 10.15.89.191 跳板机就是登录节点的ip和port  
User changwx  
Port 22112
- Host 'changwx-debug':  
HostName sist\_gpu53 debug节点每次申请完卡都要配置  
User changwx  
Port 22 这里一定要是22  
ProxyCommand C:\Windows\System32\OpenSSH\ssh.exe -W %h:%p changwx-login01
- Host 'changwx-3080':  
HostName 10.19.126.215  
User changwx  
Port 22

A yellow box highlights the 'ProxyCommand' line for the 'changwx-debug' host, and a pink box highlights the 'Port 22' line for the same host. A pink box also highlights the 'HostName' line for the 'changwx-3080' host.

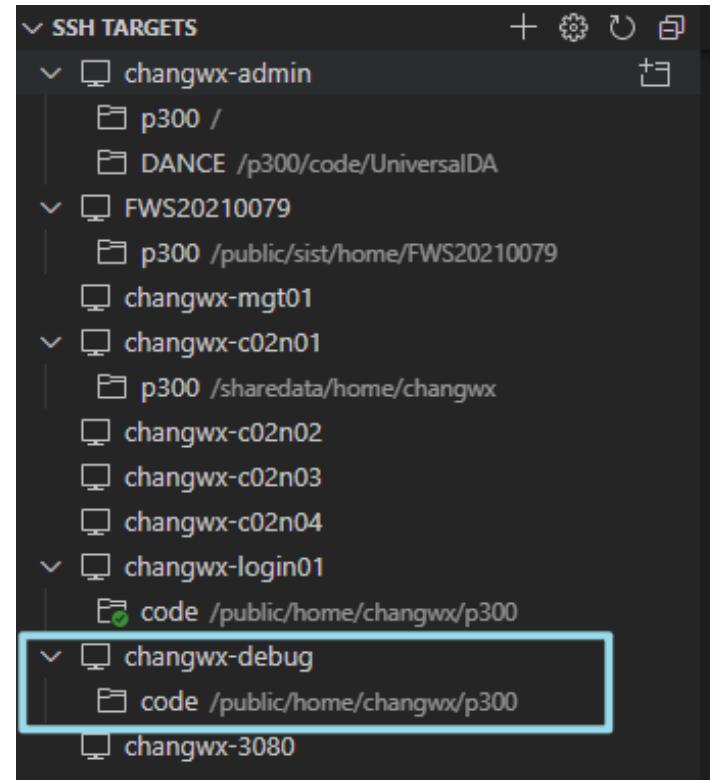
# VSCODE slurm 断点调试



上海科技大学  
ShanghaiTech University

\*例子以username为changwx展示

(c) 登录上一步申请到的计算节点（如sist\_gpu53），输入两次密码后就可以连进去了，可以直接用debug跑



立志成才报国裕民

# VSCODE slurm 断点调试



## (d) 配置debug文件

```
31
32
33     "name": "launch",
34     "type": "python",
35     "request": "launch",
36     "program": "${file}",
37     "console": "integratedTerminal",
38     "cwd": "${fileDirname}", //added by myself
39     "args": [
40         "--source", "Product",
41         "--target", "Art",
42         "--dataset", "officehome",
43         // "--dataset", "visda",
44         // "--source", "Syn",
45         // "--target", "Real",
46         // "--dataset", "domainnet",
47         // "--source", "painting",
48         // "--target", "real",
49         // "--model_path", "/public/home/changwx/p300/code/my/eccv_v2/log/ma
50         // "--model_path", "/p300/code/my/cvpr_simsiam_pretrain/log/100epoch
51         // "--amp", "True",
52         // "--da_mode", "osda",
53         // "--topk", "3",
54         // "--gpu", "0",
55         // "--test_interval", "10000"
56     ]
57 }
58 ]
59 ]
60 }
```

arg就可以写你传给你py程序的参数

立志成才报国裕民

# VSCode slurm 断点调试



### (e) 开始断点调试

```
my > eccv_v1 > main_ova_swap_cs280.py ...
  gpu_switch = False
  # device = torch.device('cpu')
else:
    import os
    os.environ["CUDA_VISIBLE_DEVICES"] = parser_args.gpu_index
    gpu_ids = list(map(int, parser_args.gpu_index))
    gpu_switch = True
    # device = torch.device('cuda')

log_dir = f'{log_path}'
print(f'log_dir is {log_dir}')
logger = SummaryWriter(log_dir)

# ======define K(for k-means)
# K = len(target_classes)
# ======define network
temperature = parser_args.temp
```

(a) 打开一个程序

main\_v6\_uot\_ti2scds\_B.py - code (SSH: changwx-debug) - Visual Studio Code

运行和调试 launch ... config.py main\_v4\_uot\_crossda.py main\_v5\_uot\_ti2scds\_ova.py ! unida-visda-config.yaml \$ main\_v6\_uot\_ti2scds\_B\_mu0.5.2.sh main\_v6\_uot\_ti2scds\_B.py

变量区

Locals

- > special variables
- > function variables
- > class variables
- EPSILON: 1e-20
- > F: <module 'torch.nn.functional' from '/...'
- > Image: <module 'PIL.Image' from '/public...'
- > ImageFilter: <module 'PIL.ImageFilter' f...
- N\_cluster\_head: 150
- > SCORERS: {'explained\_variance': make\_sco...

调用堆栈

MainThread BREAKPOINT 已暂停

<module> main\_v6\_uot\_ti2scds\_B.py 148:1

Thread-7 已暂停

监视

pi: NameError: name 'pi' is not defined

M: NameError: name 'M' is not defined

high\_conf\_label\_id: NameError: name 'high\_...

129 [optimizer feature\_optimizer\_data\_parallel(

130 > special variables

131 > function variables

132 > T\_destination: ~T\_destination

133 > device\_ids: [0]

134 dim: 0

135 dump\_patches: False

136 > module: ProtoCLSC(

137 > output\_device: 0

138 > src\_device\_obj: device(type='cuda', index=0)

139 > training: True

140 > \_apply: <bound method Module.\_apply of DataParallel(

141 > backward\_hooks: OrderedDict([])

142 > buffers: OrderedDict([])

143 > call\_impl: <bound method Module.\_call\_impl of DataParallel(

144 > forward\_hooks: OrderedDict([])

145 > forward\_pre\_hooks: OrderedDict([])

146 > get\_backward\_hooks: <bound method Module.\_get\_backward\_hooks of DataParal...

147 > get\_name: <bound method Module.\_get\_name of DataParallel(

148 > is\_full\_backward\_hook: None

optimizer\_clsopen, optimizer\_predhead, opt\_level="0"

鼠标停留在这个变量上，就会出现详细的变量信息框

ep, initial\_lr, gamma=10, power=0.75, max\_iter)

uler)

uler)

eduler)

cheduler)

(True)

按住 Alt 键可切换到编译器语言停靠

cluster\_head = nn.DataParallel(cluster\_head).train(True)

classifier\_open = nn.DataParallel(classifier\_open).train(True)

prediction\_head = nn.DataParallel(prediction\_head).train(True)

cluster\_head\_s = nn.DataParallel(cluster\_head\_s).train(True)



上海科技大学  
ShanghaiTech University

# The End



立志成才报国裕民