

## Paramiko - 服务器主机工作流

一个典型的工作过程：在主机上编写in文件，通过ssh连接到服务器后将in文件上传到服务器上，服务器进行计算，计算结束后将输出文件下载到主机上进行进一步分析。

Paramiko库实现了SSHv2协议，使得可以在Python中直接通过SSH协议对服务器进行操作，包括执行shell命令、修改文件、上传/下载文件等一系列的操作，尤其是需要进行参数化分析，或者当预处理、后处理等一些列操作可以在Python中完成时，借助这个库可以构建一整套完备的工作流..

（在jupyterlab中配合`elyra`可以实现得更加好看.）

### 1. 连接服务器

清华的探索100集群机通过公钥 - 私钥 - 私钥密码的方式登录系统，

用`paramiko`实现Python内的ssh功能中介绍了不同方式登录的paramiko接口，下面以基于公钥密钥的 SSHClient 方式登录为例

```
# 建立一个sshclient对象
ssh = paramiko.SSHClient()

# 允许将信任的主机自动加入到host_allow 列表
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

# 指定本地的RSA私钥文件
pkey = paramiko.RSAKey.from_private_key_file(
    r"C:\Users\shen\Desktop\杂物\探索100\KEY\caoby", password="it's a secret"
)

# 调用connect方法连接服务器
ssh.connect(hostname="xxx.xxx.xxx.xx", port=22, username="caoby", pkey=pkey)

# 执行命令，查看目录WORK1/shen下的内容
stdin, stdout, stderr = ssh.exec_command("cd WORK1/shen;ls", get_pty=True)
print(stdout.read().decode())
ssh.close()
```

输出如下：

10c	10c1000SKS	10c500TraPPE	4step	helloworld.txt	mdtrial
10c1000	10c500	16c	compass	log.lammps	

## 2. 上传/下载文件

通过sftp和trans进行文件传输，（trans搭建了连接的通道，sftp实现了文件传输协议，可以这样理解嘛？）

```
# 指定本地的RSA私钥文件

pkey = paramiko.RSAKey.from_private_key_file(r"C:\Users\shen\Desktop\杂物\探索100\KEY\caoby", pass

# 建立连接

trans = paramiko.Transport(('xxx.xxx.xxx.xx', 22))

trans.connect(username='caoby', pkey=pkey)

# 实例化一个 sftp对象,指定连接的通道

sftp = paramiko.SFTPClient.from_transport(trans)

# 发送文件

sftp.put(localpath='C:\\Users\\shen\\DeskTop\\helloworld.txt', remotepath='WORK1/shen/helloworld.txt')

# 下载文件

sftp.get('WORK1/shen/helloworld.txt', 'C:\\Users\\shen\\DeskTop\\new_helloworld.txt')

# 通过transport搭建的通道也可以用于执行shell命令:

# 将sshclient的对象的transport指定为以上的trans

ssh = paramiko.SSHClient()
ssh._transport = trans

# 执行命令, 和传统方法一样

stdin, stdout, stderr = ssh.exec_command('ls')
print(stdout.read().decode())

# 关闭连接

trans.close()
```

上面的代码实现了把本地 'C:\Users\shen\Desktop\helloworld.txt' 这个文件传输到服务器上 'WORK1/shen/helloworld.txt' 这个文件，接着又把这个文件下载回来，重命名为 new\_helloworld.txt.

### 3. 修改服务器中文件内容

可以使用sftp对象的open方法，这和python本身的open方法的接口是一致的。

```
f = sftp.open('WORK1/shen/test.txt','w')
f.write('hello,world')
f.close()
```

### 4. 关于环境变量

SSHclient.exec\_command默认并不是以“login”模式执行命令的，因此有些文件并不会被source（比如.bash\_profile），因此某些程序的初始化文件并不会执行，环境变量也不会被配置，这时可以将命令用“bash -lc 'some command'”封装，这时会以“login”模式执行后面的命令，-l表示login模式，-c表示将字符串中的内容当作指令，可以通过在终端中运行 `man bash` 查询各参数的含义。比如典型的 `bjobs` 命令：

```
stdin, stdout, stderr = ssh.exec_command("bash -c 'bjobs'",get_pty=True)
print(stdout.read().decode())
```

```
bash: bjobs: command not found
```

```
stdin, stdout, stderr = ssh.exec_command("bash -lc 'bjobs'",get_pty=True)
print(stdout.read().decode())
```

```
=====ATTENTION=====
=====This machine is for login only=====
=====DO NOT RUN ANY JOB ON THIS MACHINE=====
=====Otherwise will be complained by all users=====
=====ATTENTION=====
Disk quotas for user caoby (uid 798):
      Filesystem  kbytes  quota  limit  grace  files  quota
limit  grace
      /work1 799459980 1024000000      0      - 73719
0      0      -
No unfinished job found
```