

# batchRun用户手册

**Product Name :** batchRun

**Product Version :** 1.0

**Release Date :** 2022.12.12

**Contact :**  李艳青 ([liyanqing.1987@bytedance.com](mailto:liyanqing.1987@bytedance.com))

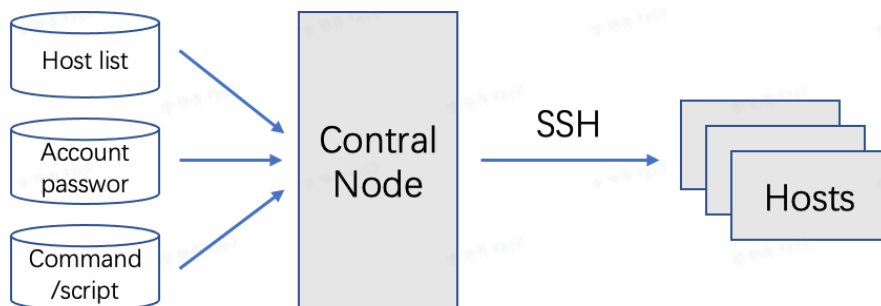
## 修订记录

日期	版本	描述	主要变更
2022.12.12	1.0	初版	无

## 一、简介

batchRun是一款适用于集成电路设计Linux环境的IT运维自动化工具，基于python语言开发，底层基于ssh实现多服务器登录管理，分布式操作，无需客户端。

batchRun的工作机制如下所示。



工具的输入信息包括三类：

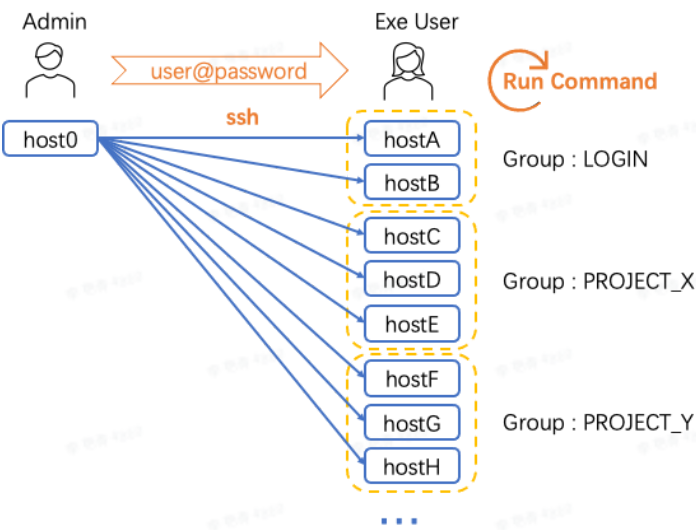
**Host list:** 机器列表。

**Account password:** 账号密码，如果是采用ssh免密或者自带加密密码系统，则无需指定自动获取。

**Command/script:** 要执行的命令或者脚本，既可以是系统命令或者普通脚本，也可以是batchRun自带脚本库中的脚本。

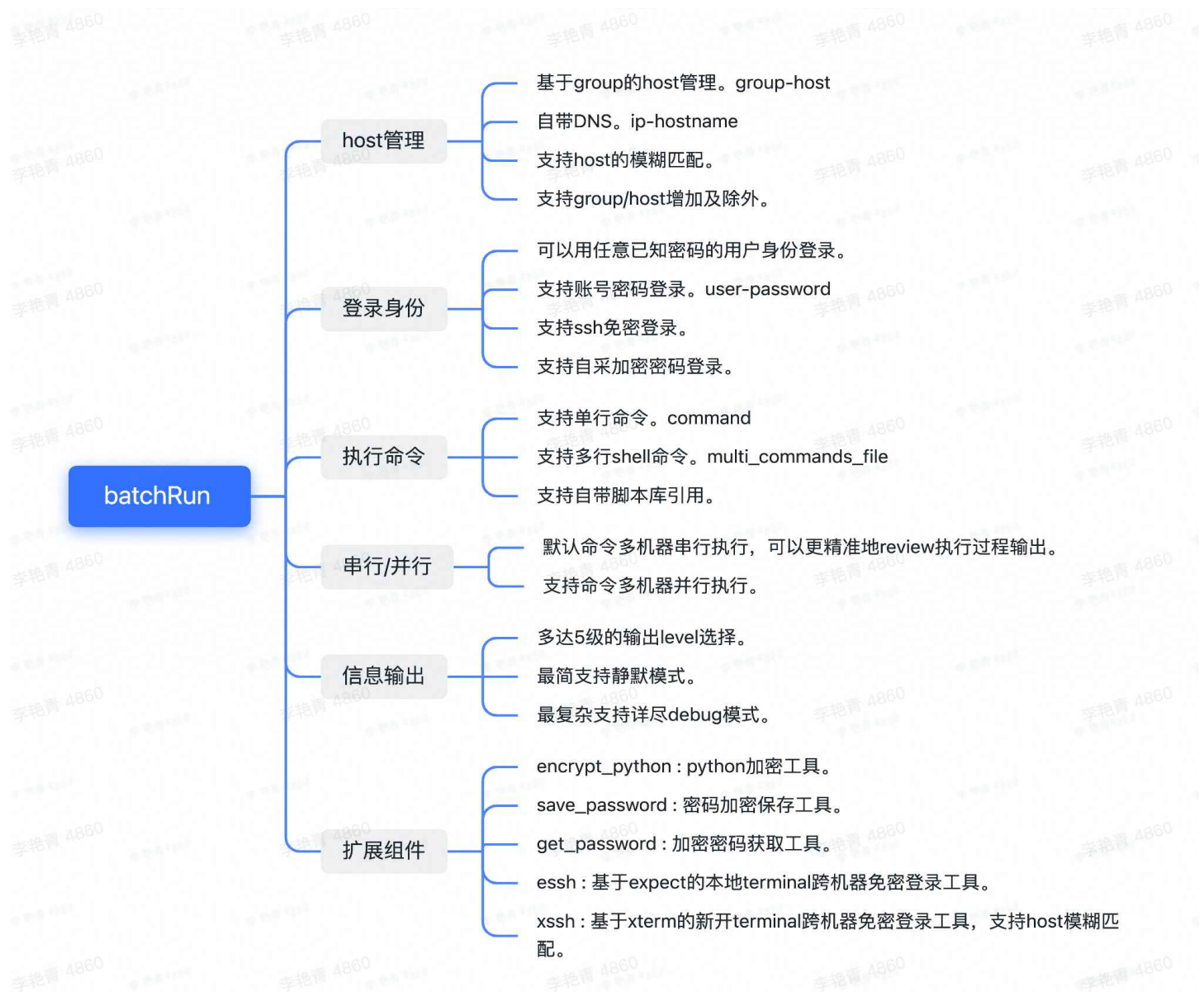
在控制节点上，batchRun基于ssh协议登录要操作的机器，在远程机器上执行指定的命令或脚本，并获取相应的输出信息。

其具体的执行流程如下图所示。



batchRun的主要目标是IC设计环境管理辅助，包括但不限于批量安装和配置操作系统、软件部署、系统信息获取等。所以batchRun并不追求Ansible之类工具的多平台兼容和强大功能，而主要关注如何更好满足IC设计环境管理需求。

batchRun的特点如下：



另外，batchRun的一个突出优点是，命令所见即所得。

IC IT管理员的一般工作习惯是，在服务器上执行shell命令（或脚本）完成配置任务，batchRun只需要将同一命令直接推送到其它服务器即可，而Ansible往往还需要用户专门配置专门语法格式的playbook才行。batchRun的操作相比较而言更加简单易操作，不需要改变IT管理员的使用习惯。

## 二、环境依赖

### 2.1 操作系统依赖

batchRun的开发和测试操作系统为**CentOS Linux release 7.9.2009 (Core)**，这也是IC设计常用的操作系统版本之一。

centos6/centos7/centos8，及对应的redhat版本应该都可以运行，主要的潜在风险在于系统库版本差异可能会影响部分组件的运行。

建议在centos7.9操作系统下使用。

## 2.2 python版本依赖

batchRun基于python开发，其开发和测试的python版本为**python3.8.8**。

不同版本的python可能会有python库版本问题，按照系统要求安装对应版本的python库即可解决。

## 2.3 共享存储

batchRun本身是分布式运行，并不依赖共享存储，但是batchRun自带脚本库，如果用batchRun运行自带脚本库中的脚本，为了确保所有执行机器上脚本可访问，需要将batchRun安装到共享存储，并在所有执行机上挂载同一个共享存储。

# 三、工具安装及配置

## 3.1 工具安装

工具安装之前，首先参照第二章“环境依赖”满足batchRun的环境依赖关系。

将安装包拷贝到安装目录，并给与合适的目录名。

```
1 [liyanqing.1987@n212-206-194 tools]$ cp -rf /ic/software/cad_tools/it/batchRun.i
```

安装包下的文件和目录如下。

```
1 [liyanqing.1987@n212-206-194 tools]$ cd batchRun/  
2 [liyanqing.1987@n212-206-194 batchRun]$ ls  
3 bin common config data install.py patch README requirements.txt scripts
```

确认python版本正确，并基于安装包中的requirements.txt安装python依赖库。（可能需要root权限）

```
1 [root@ic-admin1 batchRun]# which python3
2 /ic/software/tools/python3/3.8.8/bin/python3
3 [root@ic-admin1 batchRun]#
4 [root@ic-admin1 batchRun]# pip3 install -r requirements.txt
5 Looking in indexes: https://bytedpypi.byted.org/simple/
6 Requirement already satisfied: pexpect==4.8.0 in /ic/software/tools/python3/3.8.
7 Requirement already satisfied: ptyprocess>=0.5 in /ic/software/tools/python3/3.8
```

在安装目录下，使用命令“python3 install.py”安装batchRun。

```
1 [root@ic-admin1 batchRun]# python3 install.py
2 >>> Check python version.
3     Required python version : (3, 8)
4     Current python version : (3, 8)
5
6 >>> Generate script "/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run".
7 >>> Generate script "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/encrypt_py
8 >>> Generate script "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/get_passwo
9 >>> Generate script "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/save_passw
10 >>> Generate script "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/xssh".
11 >>> Generate config file "/ic/data/usr/liyanqing.1987/tools/batchRun/config/conf
12
13 Done, Please enjoy it.
```

## 3.2 工具配置

主要的配置文件位于 安装目录下的config目录中。

### 3.2.1 config.py

用于配置工具的一些基本设置和验证规则。

安装后默认配置如下，一般采用默认设置即可。

```
1 # Specify host list, default is "host.list" on current configure directory.
2 HOST_LIST = '/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list'
3
4 # Set log directory.
```

```
5 LOG_DIR = '/ic/data/usr/liyanqing.1987/tools/batchRun/data/log'
6
7 # Default ssh command.
8 DEFAULT_SSH_COMMAND = "ssh -XY -o StrictHostKeyChecking=no"
9
10 # Support host_ip fuzzy matching, could be "True" or "False".
11 FUZZY_MATCH = True
12
13 # Define timeout for ssh command, unit is "second".
14 TIMEOUT = 10
```

**HOST\_LIST**：指定host.list文件位置。

**LOG\_DIR**：指定batch\_run执行日志文件路径。

**DEFAULT\_SSH\_COMMAND**：指定ssh操作的命令，一般采用推荐值即可。

**FUZZY\_MATCH**：控制host模糊匹配开关，默认关闭，但是建议打开。

**TIMEOUT**：指定ssh命令的超时时间，默认为10秒。

### 3.2.2 host.list

host.list用于定义所有的机器及其分组信息，支持多种详细或者简略的格式。

建议采用“ip hostname <port>”的样式，这样还能起到默认DNS的作用。

```
1 ##### Format #####
2 # # Description
3 # [group_name]
4 # ip1
5 # ip2 <port2>
6 # ip3 <hostname3>
7 # ip4 <hostname4> <port4>
8 # sub_group
9 #####
10
11 ...
12
13 # Login servers
14 [LOGIN]
15 ETX VNC
16
17
18 # ETX login servers
```

```
19 [ETX]
20 10.212.206.195 cmp195
21 10.212.206.196 cmp196
22 10.212.206.207 cmp207
23 10.212.206.218 cmp218
24
25
26 # VNC login servers
27 [VNC]
28 10.212.206.225 cmp225
29 10.212.206.235 cmp235
30
31 ...
```

### 3.2.3 password.encrypted

这个文件由自带工具save\_password生成，无需手工配置。

建议通过save\_password将batch\_run常用用户的密码都存放进去，当前用户已经配置了ssh免密登录则无需采用这种加密密码登录的方式。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ tools/save_password -u liyanqing.1987 -p
2 [liyanqing.1987@n212-206-194 batchRun]$ cat config/password.encrypted
3 liyanqing.1987 46953b705ab979d47b81b6a5ee8fa0c7
```

## 四、工具使用

### 4.1 工具载入

batchRun的主程序是batch\_run，位于batchRun安装目录下的bin/batch\_run，安装后可以直接引用。

如果配置了modules，则可以通过module load的方式引用。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ module load cad
2 [liyanqing.1987@n212-206-194 batchRun]$ which batch_run
3 /ic/software/cad_tools/bin/batch_run
```



## 4.2 帮助信息

batch\_run的帮助信息如下。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ batch_run -h
2 usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G HOST_GROUPS [HOST_GROUPS ...]
3                      [-t TIMEOUT] [-o {0,1,2,3,4}] [--list_hosts LIST_HOSTS [LIST
4
5 optional arguments:
6   -h, --help            show this help message and exit
7   -H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
8                       Specify the host(s), could be host_ip or host_name, form
9                       "ALL" means all hosts on /ic/software/cad_tools/it/batch
10                      "~<HOST>" means exclude the specified host.
11   -G HOST_GROUPS [HOST_GROUPS ...], --host_groups HOST_GROUPS [HOST_GROUPS ...]
12                      Specify host group(s) which are on /ic/software/cad_tool
13                      "ALL" means all groups on /ic/software/cad_tools/it/batc
14                      "~<GROUP>" means exclude the specified group.
15   -P, --parallel        Run command parallel on specified host(s), default is in
16   -u USER, --user USER Specify the user name when connecting host as.
17   -p PASSWORD, --password PASSWORD
18                       Specify the user password when connecting host with.
19   -c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]
20                       Specify command you want run on specified host(s).
21   -m MULTI_COMMANDS_FILE, --multi_commands_file MULTI_COMMANDS_FILE
22                       Specify a script with multi commands, will execute line
23   -t TIMEOUT, --timeout TIMEOUT
24                       Specify ssh command timeout, default is 10 seconds.
25   -o {0,1,2,3,4}, --output_message_level {0,1,2,3,4}
26                       Specify output message level, default is "3".
27                       "0" means silence;
28                       "1" only show host info;
29                       "2" only show one line output message;
30                       "3" show normal output message;
31                       "4" show verbose output message.
32   --list_hosts LIST_HOSTS [LIST_HOSTS ...]
33                       List all or specified-group hosts on /ic/software/cad_to
34                       "all" or "ALL" means all hosts on host list file.
35                       "<group>" means hosts on specified groups.
```

**--help** : 打印帮助信息。



- hosts** : 指定机器列表，可以是机器ip，也可以是机器名。如果是“ALL”，则意味着配置文件host.list中所有的机器。
- host\_groups** : 指定机器组。如果是“ALL”，则意味着配置文件host.list中所有的机器。
- parallel** : 指定并行运行模式，默认是串行。如果希望获取有规律的输出信息，建议串行。
- user** : 指定ssh登录远程机器时的用户，默认是当前用户。
- password** : 指定ssh登录远程机器时的用户密码。如果配置了ssh免密登录，或者加密密码已经被保存到了password.encrypted中，此处无需指定。
- command** : 到远程机器上要执行的命令。可以是系统命令或者指定脚本，也可以是batchRun自带脚本库中的脚本。
- multi\_commands\_files** : 如果ssh登录到远程机器上需要执行多行命令，可以将多行命令写入到文件中，然后通过--multi\_commands\_files来指定多命令文本。
- timeout** : 指定ssh的超时时间，默认为10秒。
- output\_message\_level** : 指定执行命令时输出信息的详细程度，分为0-4共五个登记，其中“0”意味着完全静默，“4”意味着最详细的信息输出。
- list\_hosts** : 列出配置文件host.list中所有的机器。

## 4.3 功能介绍

### 4.3.1 列出预设机器

如果不清楚batch\_run预设的机器及机器组设置，可以使用“batch\_run --list\_hosts all”查看。

```
1 [root@ic-admin1 batchRun]# batch_run --list_hosts all
2 GROUP : SERVICE
3      10.212.240.5      sftp-upload      5566
4      10.212.240.6      sftp-download    5566
5      10.212.240.12     ic-idm-01
6 ...
7 GROUP : PHYSICAL
8      LOGIN/
9      COMPUTING/
```

如果已知机器组，也可以使用“batch\_run --list\_hosts <GROUP>”查看指定机器组的具体信息。

```
1 [root@ic-admin1 batchRun]# batch_run --list_hosts ETX
```

```
2 GROUP : ETX
3      10.212.206.194      n212-206-194
4      10.212.206.195      n212-206-195
5      10.212.206.196      n212-206-196
6      10.212.206.207      n212-206-207
7      10.212.206.218      n212-206-218
8      10.212.204.130      n212-204-130
```

### 4.3.2 指定用户名和密码登录远程机器

可以采用指定用户名和密码的方式登录任意机器执行指定命令，操作方式为“`batch_run -u <USER> -p <PASSWORD>`”。

```
1 [root@ic-admin1 batchRun]# batch_run -u ic_admin -p *** -c "cat /etc/redhat-release"
2
3 >>> 10.212.207.130
4      CentOS Linux release 7.6.1810 (Core)
5
6 >>> 10.212.207.136
7      CentOS Linux release 7.6.1810 (Core)
8
9 >>> 10.212.207.137
10     CentOS Linux release 7.6.1810 (Core)
11
12 >>> 10.212.207.138
13     CentOS Linux release 7.6.1810 (Core)
```

因为明文密码会导致敏感信息泄露，所以并不推荐这种方式，建议配置ssh免密登录，或者采用内置加密密码的方式登录远程机器。

### 4.3.3 采用内置加密密码登录远程机器

使用内置工具save\_password为ic\_admin账号保存加密密码。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ tools/save_password -u ic_admin -p ***
```

其储存的密码为加密样式。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ cat config/password.encrypted
2 ic_admin  adf95ce69c9e3cf62abe5d1460fb6b8a
```

此时，batch\_run用ic\_admin账号登录远程机器时，可以使用这个加密密码，以防止明文导致密码泄露的问题。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -c "cat /etc/redhat-release" -G PALLADIUM
2
3 >>> 10.212.207.130
4     CentOS Linux release 7.6.1810 (Core)
5
6 >>> 10.212.207.136
7     CentOS Linux release 7.6.1810 (Core)
8
9 >>> 10.212.207.137
10    CentOS Linux release 7.6.1810 (Core)
11
12 >>> 10.212.207.138
13    CentOS Linux release 7.6.1810 (Core)
```

请注意，一定要以密码登陆的方式成为ic\_admin用户，否则batch\_run会无法判断当前实际用户是否知悉ic\_admin账号密码（比如root账号su成ic\_admin，并不需要知道ic\_admin的密码），从而导致拒绝执行命令。

#### 4.3.4 指定机器执行命令

batch\_run支持用“batch\_run -H <HOST> <HOST> ...”的方式执行机器，HOST既可以是ip地址，也可以是hostname（hostname一定要在/etc/hosts或者host.list中配置过）。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -c "cat /etc/redhat-release" -H 10.212.207.130
2
3 >>> 10.212.207.130
4     CentOS Linux release 7.6.1810 (Core)
5 [ic_admin@ic-admin1 ~]$
6 [ic_admin@ic-admin1 ~]$ batch_run -c "cat /etc/redhat-release" -H n212-207-130
```

```
7
8 >>> n212-207-130
9     CentOS Linux release 7.6.1810 (Core)
```

### 4.3.5 机器模糊匹配

同时，batch\_run也支持ip地址或者hostname的模糊匹配，如果匹配上多项，则多项同时执行。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -c "cat /etc/redhat-release" -H 130
2 [FUZZY MATCH] 130 -> 10.212.204.130
3 [FUZZY MATCH] 130 -> 10.212.207.130
4
5 >>> 10.212.204.130
6     Warning: Permanently added '10.212.204.130' (ECDSA) to the list of known hos
7     CentOS Linux release 7.9.2009 (Core)
8
9 >>> 10.212.207.130
10    CentOS Linux release 7.6.1810 (Core)
```

### 4.3.6 串行执行和并行执行

先创建一个要执行的脚本。

```
1 [ic_admin@ic-admin1 ~]$ cat /home/ic_admin/touch_hostname.sh
2 #!/bin/bash
3
4 echo `hostname`
5 touch `hostname`
```

默认是串行执行，效果如下。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -c /home/ic_admin/touch_hostname.sh -G ETX -o
2 >>> 10.212.206.194      n212-206-194
3 >>> 10.212.206.195      n212-206-195
4 >>> 10.212.206.196      n212-206-196
5 >>> 10.212.206.207      n212-206-207
```

```

6 >>> 10.212.206.218      n212-206-218
7 >>> 10.212.204.130      n212-204-130
8 [ic_admin@ic-admin1 ~]$
9 [ic_admin@ic-admin1 ~]$ ls
10 n212-204-130  n212-206-194  n212-206-195  n212-206-196  n212-206-207  n212-206-2

```

我们看到，脚本在指定机器上串行执行，明确打印output信息，并生成了对应文件。

下面我们删掉生成的文件，用并行方式再执行一遍。

并行命令为“batch\_run -P”。

```

1 [ic_admin@ic-admin1 ~]$ rm -rf n212-20*
2 [ic_admin@ic-admin1 ~]$ batch_run -c /home/ic_admin/touch_hostname.sh -G ETX -o
3 >>> 10.212.206.195 >>> 10.212.206.207 >>> 10.212.206.194 >>> 10.212.206.218 >>>
4     n212-206-194
5     n212-206-196
6     n212-206-195
7     n212-206-207
8     n212-204-130
9 [ic_admin@ic-admin1 ~]$
10 [ic_admin@ic-admin1 ~]$ ls
11 n212-204-130  n212-206-194  n212-206-195  n212-206-196  n212-206-207  n212-206-2

```

我们看到文件正常生成，说明脚本在每台机器上都正常执行了，但是没有任何output信息，这是因为并行模式下输出信息是乱序的，和机器对应不上，因此batch\_run在并行模式下什么信息都不会输出！

### 4.3.7 执行多行命令

如果在同一台机器上需要执行多行命令，可以采用参数“batch\_run -m <FILE>”来实现。

我们先创建一个文件，写入多行命令。（最好每一行都是有效命令行）

```

1 [ic_admin@ic-admin1 ~]$ cat get_host_info
2 cat /etc/redhat-release
3 cat /proc/cpuinfo | grep name | sort | uniq
4 cat /proc/meminfo | grep MemTotal

```

执行多行命令。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -m get_host_info.sh
2
3 >>> 10.212.207.130
4     CentOS Linux release 7.6.1810 (Core)
5     model name       : Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz
6     MemTotal:        1056265940 kB
```

如果远程机器挂载了共享目录，可以用脚本的方式取代多行模式。但是在远程机器没有共享目录的情况下，是没办法执行指定脚本的，只能以多行模式来实现同一效果。

### 4.3.8 输出信息层级

batch\_run的output message共有5个层级可选，效果分别如下。

- “0”级输出，静默模式，无任何输出。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c "lsb_release -a" -o 0
2 [ic_admin@ic-admin1 ~]$
```

- “1”级输出，极简模式，仅显示执行的机器。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c "lsb_release -a" -o 1
2 >>> 10.212.207.130
```

- “2”级输出，当行模式，仅显示output message的第一行，且显示到机器信息的同一行。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c "lsb_release -a" -o 2
2 >>> 10.212.207.130     LSB Version:          :core-4.1-amd64:core-4.1-noarch:cxx-4
```

- “3” 级输出，正常模式，正常显示output message。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c "lsb_release -a" -o 3
2
3 >>> 10.212.207.130
4     LSB Version:           :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-no
5     Distributor ID:        CentOS
6     Description:           CentOS Linux release 7.6.1810 (Core)
7     Release:               7.6.1810
8     Codename:              Core
```

- “4” 级输出，详细模式，除正常output message外，会打印额外的debug信息。

```
1 [ic_admin@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c "lsb_release -a" -o 4
2
3 >>> 10.212.207.130
4     /ic/software/cad_tools/it/batchRun/tools/run_ssh_command.py -c "ssh -XY -o S
5     ==== output ====
6     LSB Version:           :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-no
7     Distributor ID:        CentOS
8     Description:           CentOS Linux release 7.6.1810 (Core)
9     Release:               7.6.1810
10    Codename:              Core
11    =====
```

默认的输出级别是“3”。

## 五、辅助工具

出于安全和便捷性的考虑，batch\_run自带一些辅助工具。

### 5.1 save\_password

用于加密用于密码并保存。



```
1 [ic_admin@cmp195 tools]$ ./save_password -h
2 usage: save_password.py [-h] [-P PASSWORD_FILE] [-u USER] -p PASSWORD
3
4 optional arguments:
5   -h, --help            show this help message and exit
6   -P PASSWORD_FILE, --password_file PASSWORD_FILE
7                           Specify the user password file, default is "/ic/software/cad_tools/it/batch_run/config/password.encrypted".
8   -u USER, --user USER  Specify user name, default is current user.
9   -p PASSWORD, --password PASSWORD
10                          Specify user password.
```

-P PASSWORD\_FILE, 用于指定保存账号密码信息的文件，一般无需指定，采用默认值即可。

-u USER, 指定用户名，默认是当前用户。

-p PASSWORD, 指定用户密码。

用户密码被加密保存后，在工具引用时可以直接读取，无需明文输入，从而保证了密码安全性。

## 5.2 get\_password

用于查看自己保存的密码，注意，只能查看用户自己的密码。

```
1 [ic_admin@cmp195 tools]$ ./get_password -h
2 usage: get_password.py [-h] [-P PASSWORD_FILE] [-u USER]
3
4 optional arguments:
5   -h, --help            show this help message and exit
6   -P PASSWORD_FILE, --password_file PASSWORD_FILE
7                           Specify the user password file, default is "/ic/software/cad_tools/it/batch_run/config/password.encrypted".
8   -u USER, --user USER  Specify the user name, default is current user.
```

-P PASSWORD\_FILE, 用于指定保存账号密码信息的文件，一般无需指定，采用默认值即可。

-u USER, 指定用户名，默认是当前用户。

## 5.3 essh

采用expect的方式无密码登录指定机器。

如果用save\_password保存过加密密码，登录机器不依赖用户的ssh免密登录，也不需要输入密码，比较方便。

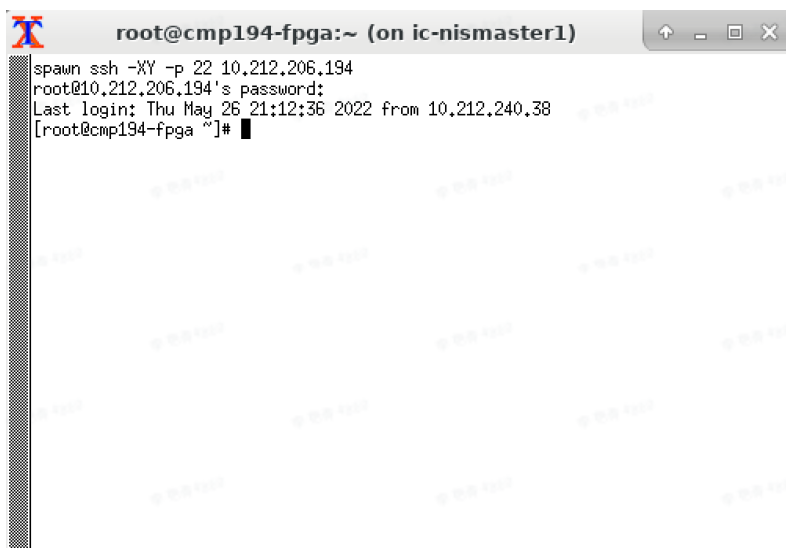
```
1 [root@ic-nismaster1 tools]# ./essh 10.212.206.194
2 spawn ssh -XY -p 22 10.212.206.194
3 root@10.212.206.194's password:
4 Last login: Sat May 21 13:28:59 2022 from 10.212.206.207
5 [root@cmp194-fpga ~]#
```

## 5.4 xssh

采用xterm的方式无密码登录指定机器，开启模糊匹配的前提下，机器支持简写。

如果用save\_password保存过加密密码，登录机器不依赖用户的ssh免密登录，也不需要输入密码，而且还支持host简写，非常方便，强烈推荐！

```
1 [root@ic-nismaster1 tools]# xssh 194
2 [FUZZY MATCH] 194 -> 10.212.206.194
3 xterm -e "/ic/software/cad_tools/it/batch_run/tools/essh 10.212.206.194 22" &
```



如果有多个匹配的hosts，会列出供用户选择。

```
1 [root@ic-nismaster1 tools]# xssh 130
2 [FUZZY MATCH] 130 -> 10.212.204.130
3 [FUZZY MATCH] 130 -> 10.212.207.130
4 Below are possible hosts:
5 [0] 10.212.204.130
6 [1] 10.212.207.130
7 [2] None of above hosts
8
9 Please choice one (number): 1
10 xterm -e "/ic/software/cad_tools/it/batch_run/tools/essh 10.212.207.130 22" &
```



```
root@n212-207-130:~ (on ic-nismaster1)
spawn ssh -XY -p 22 10.212.207.130
root@10.212.207.130's password:
Last login: Mon May 23 16:28:15 2022 from ic-admin1
[root@n212-207-130 ~]#
```

## 六、自带脚本库

batch\_run可以执行系统命令，也可以指定指定路径下的脚本（前提是远程机器挂载了共享目录，可以读到这个脚本），有时候我们为了方便，会把一些常用操作写成脚本来执行（IT运维自动化），为了方便地引用这些脚本，可以把它们放到batchRun的自带脚本库“<INSTALL\_PATH>/scripts”下面，这样就可以无路径引用这些脚本。

比如我们在batchRun安装目录下的scripts目录中写一个脚本，用于获取服务器的基本信息。

```
1 [liyanqing.1987@n212-206-194 batchRun]$ cat scripts/get_server_info/get_server_i
2 #!/ic/software/tools/python3/3.8.8/bin/python3
3 # -*- coding: utf-8 -*-
4 ...
5 import os
6 import re
7 import socket
8
```

执行这个自带脚本库中的脚本时，无需指明路径，只写上脚本名即可。

```
1 [ic_admin@ic-admin1 scripts]$ batch_run -H 10.212.207.130 -c get_server_info.py
2
3 >>> 10.212.207.130
4     ip : 10.212.207.130
5     hostname : n212-207-130
6     os : CentOS Linux release 7.6.1810 (Core)
7     cpu : 96
8     mem : 1007G
```

这样我们就可以把如下类型的IT运维自动化脚本写入batchRun，然后便捷引用。

- 操作系统初始化。
- 操作系统优化。
- 磁盘挂载。
- 常用服务安装和启动。
- 常用系统配置。