

batchRun 用户手册

Product Name : batchRun

Product Version : V1.0

Release Date : 2023.2.12

Contact : @李艳青 (liyanqing.1987@bytedance.com)

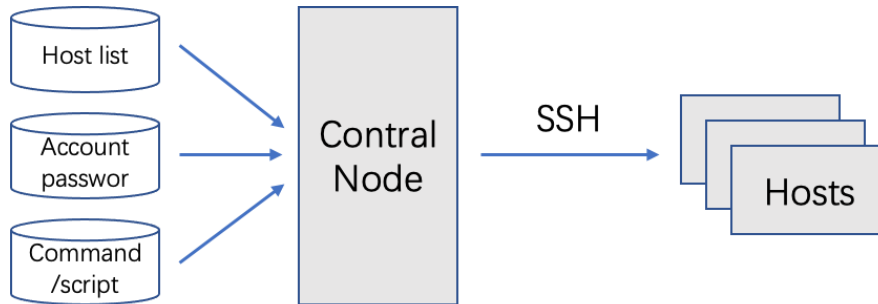
目录

一、简介.....	3
二、环境依赖.....	5
2.1 操作系统依赖	5
2.2 PYTHON 版本依赖	5
2.3 共享存储	5
三、工具安装及配置.....	6
3.1 工具下载	6
3.2 工具安装	7
3.3 工具配置	8
3.3.1 config.py	8
3.3.2 host.list	9
3.3.3 password.encrypted	10
四、工具使用.....	11
4.1 工具载入	11
4.2 帮助信息	11
4.3 功能介绍	13
4.3.1 列出预设机器	13
4.3.2 指定用户名和密码登录远程机器	14
4.3.3 采用内置加密密码登录远程机器	14
4.3.4 指定机器执行命令	15
4.3.5 指定机器组执行命令	15
4.3.6 指定 LSF 队列机器执行命令	16
4.3.7 机器模糊匹配	16
4.3.8 串行执行和并行执行	17
4.3.9 执行多行命令	18
4.3.10 批量配置 ssh 免密	19
4.3.11 输出信息层级	20
五、辅助工具.....	22
5.1 SAVE_PASSWORD	22
5.2 GET_PASSWORD	22
5.3 ESSH	23
5.4 XSSH	23
六、自带脚本库.....	25
附录.....	27
附 1. 变更历史	27

一、简介

batchRun 是 ByteDance 开源的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：

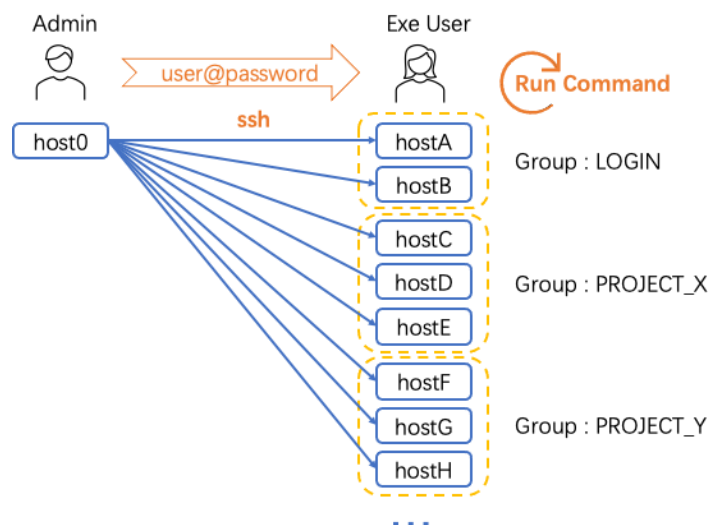
Host list: 机器列表。

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。

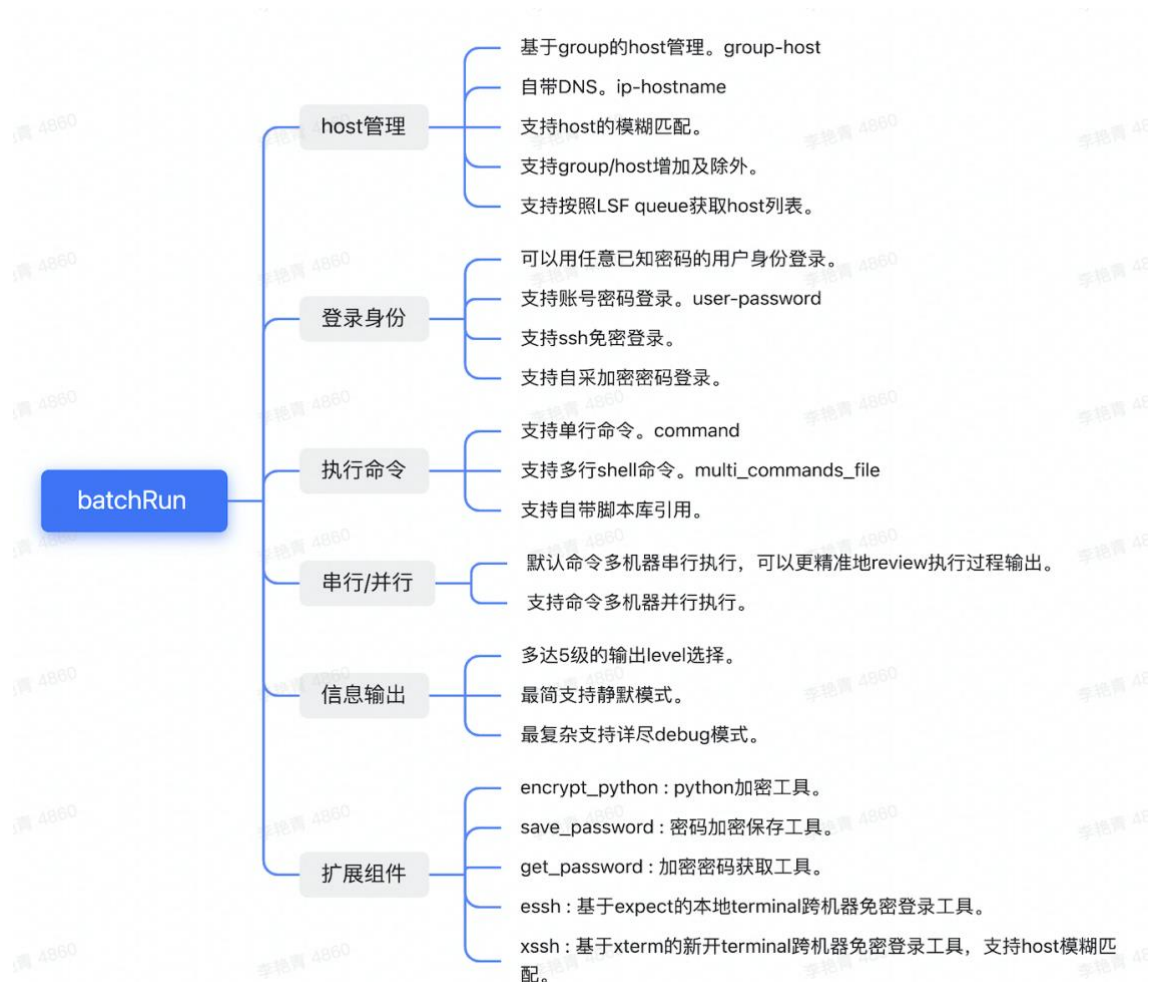
在控制节点上，batchRun 基于 ssh 协议登录要操作的机器，在远程机器上执行指定的命令或脚本，并获取相应的输出信息。

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，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（或其它脚本）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 工具下载

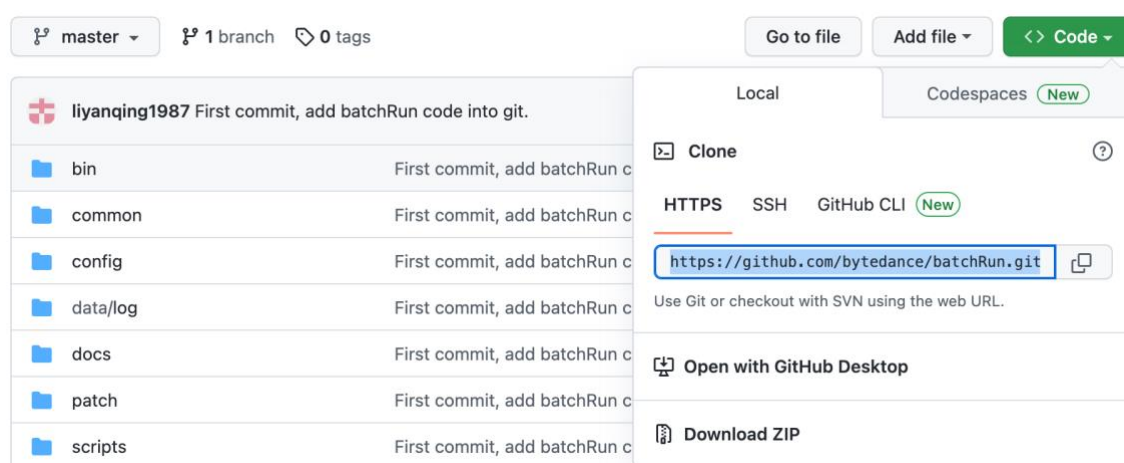
batchRun 的 github 路径位于 <https://github.com/bytedance/batchRun>。

The screenshot displays the GitHub interface for the `bytedance/batchRun` repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below this, the repository name `bytedance/batchRun` is shown as public, with options to Watch (1), Fork (0), and Star (0). A secondary navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows the `master` branch with 1 branch and 0 tags. A list of files and directories is provided, each with a commit message and timestamp (all from 2 hours ago). The files include `bin`, `common`, `config`, `data/log`, `docs`, `patch`, `scripts`, `tools`, `LICENSE.txt`, `Notice.txt`, `README`, `install.py`, and `requirements.txt`. On the right, the 'About' section describes `batchRun` as an ansible-similar IT automation system. Below this are sections for Releases (no releases published), Packages (no packages published), and Languages (Python 94.5%, Shell 5.5%).

可以采用“`git clone https://github.com/bytedance/batchRun.git`”的方式拉取源代码。

```
Bash
bytedance@C02FT5LHMD6R Downloads % git clone
https://github.com/bytedance/batchRun.git
Cloning into 'batchRun'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 32 (delta 1), reused 32 (delta 1), pack-reused 0
Receiving objects: 100% (32/32), 800.59 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

也可以在 batchRun 的 github 页面上，Code -> Download ZIP 的方式拉取代码包。



3.2 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。

安装包下的文件和目录如下。

```
Bash
[root@ic-admin1 tools]# cd batchRun/
[root@ic-admin1 batchRun]# ls
bin common config data docs install.py LICENSE.txt Notice.txt
patch README requirements.txt scripts tools
```

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)

```
Bash
[root@ic-admin1 batchRun]# which python3
/ic/software/tools/python3/3.8.8/bin/python3
[root@ic-admin1 batchRun]#
[root@ic-admin1 batchRun]# pip3 install -r requirements.txt
Looking in indexes: https://bytedpypi.byted.org/simple/
Requirement already satisfied: pexpect==4.8.0 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from -
r requirements.txt (line 1)) (4.8.0)
Requirement already satisfied: ptyprocess>=0.5 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from
```

```
pexpect==4.8.0->-r requirements.txt (line 1)) (0.7.0)
```

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

```
Bash
[root@ic-admin1 batchRun]# python3 install.py
>>> Check python version.
    Required python version : (3, 8)
    Current  python version : (3, 8)

>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/encrypt_python".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/get_password".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/save_password".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/xssh".
>>> Generate config file
"/ic/data/usr/liyanqing.1987/tools/batchRun/config/config.py".

Done, Please enjoy it.
```

3.3 工具配置

主要的配置文件位于 安装目录下的 config 目录中。

3.3.1 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。

```
Bash
# Specify host list, default is "host.list" on current configure
directory.
HOST_LIST =
'/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list'
```



```
# Set log directory.
LOG_DIR = '/ic/data/usr/liyanqing.1987/tools/batchRun/data/log'

# Default ssh command.
DEFAULT_SSH_COMMAND = "ssh -o StrictHostKeyChecking=no"

# Support host_ip fuzzy matching, could be "True" or "False".
FUZZY_MATCH = True

# Define timeout for ssh command, unit is "second".
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。

3.3.2 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。

```
C#
#### Format ####
# # Description
# [group_name]
# ip1
# ip2 <port2>
# ip3 <hostname3>
# ip4 <hostname4> <port4>
# sub_group
#####

...

# Login servers
[LOGIN]
```

```
ETX VNC
```

```
# ETX login servers
[ETX]
10.212.206.194 cmp194
10.212.206.195 cmp195
10.212.206.196 cmp196
10.212.206.207 cmp207
10.212.206.218 cmp218

# VNC login servers
[VNC]
10.212.206.225 cmp225
10.212.206.235 cmp235

...
```

3.3.3 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。

```
Bash
[root@ic-admin1 batchRun]# tools/save_password -u liyanqing.1987 -p
***
[root@ic-admin1 batchRun]# cat config/password.encrypted
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。

Bash

```
[root@ic-admin1 batchRun]# module load cad
[root@ic-admin1 batchRun]# which batch_run
/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run
```

4.2 帮助信息

batch_run 的帮助信息如下。

Bash

```
[root@ic-admin1 bin]# batch_run -h
usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G HOST_GROUPS
[HOST_GROUPS ...]] [-lsf_queues LSF_QUEUES [LSF_QUEUES ...]] [-P]
[-u USER] [-p PASSWORD] [-c COMMAND
[COMMAND ...]] [-m MULTI_COMMANDS_FILE] [-t TIMEOUT] [-o
{0,1,2,3,4}]
[--list_hosts LIST_HOSTS [LIST_HOSTS ...]]

optional arguments:
  -h, --help                show this help message and exit
  -H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
                           Specify the host(s), could be host_ip or
                           host_name, format is "host" or "host:port".
                           "ALL" means all hosts on
                           /ic/software/cad_tools/it/batchRun/config/host.list.
                           "~<HOST>" means exclude the specified host.
  -G HOST_GROUPS [HOST_GROUPS ...], --host_groups HOST_GROUPS
[HOST_GROUPS ...]
                           Specify host group(s) which are on
                           /ic/software/cad_tools/it/batchRun/config/host.list.
                           "ALL" means all groups on
                           /ic/software/cad_tools/it/batchRun/config/host.list.
                           "~<GROUP>" means exclude the specified group.
```

```

-lsf_queues LSF_QUEUES [LSF_QUEUES ...], --lsf_queues LSF_QUEUES
[LSF_QUEUES ...]
                                Get host list based on LSF queues.
                                "ALL" means all LSF hosts.
                                "~<QUEUE>" means exclude the specified queue.
-P, --parallel                  Run command parallel on specified host(s),
default is in serial.
-u USER, --user USER          Specify the user name when connecting host
as.
-p PASSWORD, --password PASSWORD
                                Specify the user password when connecting
host with.
-c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]
                                Specify command you want run on specified
host(s).
-m MULTI_COMMANDS_FILE, --multi_commands_file MULTI_COMMANDS_FILE
                                Specify a script with multi commands, will
execute line by line.
-t TIMEOUT, --timeout TIMEOUT
                                Specify ssh command timeout, default is 10
seconds.
-o {0,1,2,3,4}, --output_message_level {0,1,2,3,4}
                                Specify output message level, default is
"3".
                                "0" means silence;
                                "1" only show host info;
                                "2" only show one line output message;
                                "3" show normal output message;
                                "4" show verbose output message.
--list_hosts LIST_HOSTS [LIST_HOSTS ...]
                                List all or specified-group hosts on
/ic/software/cad_tools/it/batchRun/config/host.list.
                                "all" or "ALL" means all hosts on host list
file.
                                "<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 : 根据服务器组获取服务器列表。如果是“ALL”，则意味着配置文件 host.list 中所有的机器。

--lsf_queues : 根据 LSF 队列获取服务器列表。如果是“ALL”，则意味着 LSF 中的所有服务器。

--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。

```
Bash
[root@ic-admin1 ~]# batch_run --list_hosts all
GROUP : SERVICE
      10.212.240.5      sftp-upload      5566
      10.212.240.6      sftp-download    5566
      10.212.240.12     ic-idm-01
...
GROUP : PHYSICAL
      LOGIN/
      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。

```
Bash
[root@ic-admin1 ~]# batch_run --list_hosts ETX
GROUP : ETX
    10.212.206.194      n212-206-194
    10.212.206.195      n212-206-195
    10.212.206.196      n212-206-196
    10.212.206.207      n212-206-207
    10.212.206.218      n212-206-218
```

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>”。

```
Bash
[root@ic-admin1 ~]# batch_run -u ic_admin -p *** -H 10.212.206.194 -c whoami

>>> 10.212.206.194
    ic_admin
```

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。

```
Bash
[root@ic-admin1 ~]# tools/save_password -u ic_admin -p ***
```

其储存的密码为加密样式。

```
Bash
[root@ic-admin1 ~]# cat config/password.encrypted
```

```
ic_admin  adf95ce69c9e3cf62abe5d1460fb6b8a
```

此时，batch_run 用 ic_admin 账号登录远程机器时，可以直接使用这个加密密码，以防止明文导致密码泄露的问题。

```
Bash
[ic_admin@ic-admin1 ~]$ batch_run -G ETX -c "cat /etc/redhat-
release"

>>> 10.212.206.194
      CentOS Linux release 7.9.2009 (Core)

>>> 10.212.206.195
      CentOS Linux release 7.9.2009 (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> ...”的方式指定机器，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）。

```
Bash
[root@ic-admin1 bin]# batch_run -H 10.212.206.194 n212-206-195 -c
hostname -o 2
>>> 10.212.206.194      n212-206-194
>>> n212-206-195       n212-206-195
```

4.3.5 指定机器组执行命令

batch_run 支持用“batch_run -G <GROUP> <GROUP> ...”的方式指定机器，
GROUP 必须在 host.list 中有定义。

Bash

```
[root@ic-admin1 ~]# batch_run -G ETX -c hostname -o 2
>>> 10.212.206.194      n212-206-194
>>> 10.212.206.195      n212-206-195
>>> 10.212.206.196      n212-206-196
>>> 10.212.206.207      n212-206-207
>>> 10.212.206.218      n212-206-218
```

也可以采用除外的方式排除掉指定的 group 或者 host，比如不想要 194 和 195 两台机器，支持按照 ip/hostname 的方式排除机器。

Bash

```
[root@ic-admin1 ~]# batch_run -G ETX -H ~10.212.206.194 ~n212-206-195 -c hostname -o 2
>>> 10.212.206.196      n212-206-196
>>> 10.212.206.207      n212-206-207
>>> 10.212.206.218      n212-206-218
```

4.3.6 指定 LSF 队列机器执行命令

batch_run 还支持用“batch_run --lsf_queues <QUEUE> <QUEUE> ...”的方式指定 LSF 队列中的机器。

Bash

```
[root@ic-admin1 bin]# batch_run --lsf_queues a8-test swan -c
hostname -o 2
>>> n232-132-012      n232-132-012
>>> n232-130-194      n232-130-194
>>> n212-206-210      n212-206-210
```

4.3.7 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。

Bash

```
[root@ic-admin1 bin]# batch_run -c "cat /etc/redhat-release" -H 130
[FUZZY MATCH] 130 -> 10.212.204.130
```



```
[FUZZY MATCH] 130 -> 10.212.207.130

>>> 10.212.204.130
CentOS Linux release 7.9.2009 (Core)

>>> 10.212.207.130
CentOS Linux release 7.6.1810 (Core)
```

4.3.8 串行执行和并行执行

先创建一个要执行的脚本。

```
Bash
[root@ic-admin1 ~]# cat /ic/ic_admin/touch_hostname.sh
#!/bin/bash

echo `hostname`
touch `hostname`
```

默认是串行执行，效果如下。

```
Bash
[root@ic-admin1 ~]# batch_run -c /home/ic_admin/touch_hostname.sh -G
ETX -o 2
>>> 10.212.206.194      n212-206-194
>>> 10.212.206.195      n212-206-195
>>> 10.212.206.196      n212-206-196
>>> 10.212.206.207      n212-206-207
>>> 10.212.206.218      n212-206-218
[root@ic-admin1 ~]#
[root@ic-admin1 ~]# ls
n212-206-194  n212-206-195  n212-206-196  n212-206-207  n212-206-218
touch_hostname.sh
```

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”。

```

Bash
[root@ic-admin1 ~]# rm -rf n212-20*
[root@ic-admin1 ~]# batch_run -c /home/ic_admin/touch_hostname.sh -G
ETX -o 2 -P
>>> 10.212.206.195 >>> 10.212.206.207 >>> 10.212.206.194 >>>
10.212.206.218 >>> 10.212.206.196 >>> 10.212.204.130      n212-206-
218
      n212-206-194
      n212-206-196
      n212-206-195
      n212-206-207
      n212-204-130
[root@ic-admin1 ~]$
[root@ic-admin1 ~]$ ls
n212-206-194  n212-206-195  n212-206-196  n212-206-207  n212-206-218
touch_hostname.sh

```

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！

4.3.9 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）

```

Bash
[root@ic-admin1 ~]# cat get_host_info
cat /etc/redhat-release
cat /proc/cpuinfo | grep name | sort | uniq
cat /proc/meminfo | grep MemTotal

```

执行多行命令。

```

Bash
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -m get_host_info.sh

```

```
>>> 10.212.207.130
CentOS Linux release 7.6.1810 (Core)
model name      : Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz
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。

4.3.10 批量配置 ssh 免密

batchRun 自有脚本库位于安装目录下的 scripts 目录，可以放入自有脚本。“nopassword_ssh.sh”就是一个自带脚本，可以用于共享存储条件下配置服务器间用户 ssh 免密登录。

```
Bash
[liyanqing.1987@n212-206-194 ~]$ batch_run -G ETX -c
nopassword_ssh.sh

>>> 10.212.206.194
Generating public/private rsa key pair.
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
The key fingerprint is:
SHA256:Dd+bRsE5+prVp/7UDb16CR09ik/xW1bOuVkt1f5ctkc
liyanqing.1987@n212-206-194
The key's randomart image is:
+---[RSA 2048]-----+
|
|      . .
|      . = ..
|      + o +.++
|      S + +.B==
|      +.*oOE
|      O.**%
|      = +oB+
|      o .oo.o
+-----[SHA256]-----+

>>> 10.212.206.195
Generating public/private rsa key pair.
```

```
Your identification has been saved in id_rsa.  
Your public key has been saved in id_rsa.pub.  
...
```

配置完毕后，即可在指定服务器间免密登录。

```
Bash  
[liyanqing.1987@n212-206-194 ~]$ ssh 10.212.206.195  
Last login: Sun Jan  8 10:30:32 2023  
[liyanqing.1987@n212-206-195 ~]$
```

4.3.11 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。

```
Bash  
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -c "lsb_release -a"  
-o 0  
[root@ic-admin1 ~]#
```

- “1”级输出，极简模式，仅显示执行的机器。

```
Bash  
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -c "lsb_release -a"  
-o 1  
>>> 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。

```
Bash  
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -c "lsb_release -a"  
-o 2  
>>> 10.212.207.130      LSB Version:          :core-4.1-amd64:core-4.1-  
noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-  
noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-
```

```
amd64:printing-4.1-noarch
```

- “3”级输出，正常模式，正常显示 output message。

```
Bash
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -c "lsb_release -a"
-o 3

>>> 10.212.207.130
      LSB Version:          :core-4.1-amd64:core-4.1-noarch:cxx-4.1-
amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-noarch:languages-
4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-
noarch
      Distributor ID:      CentOS
      Description:        CentOS Linux release 7.6.1810 (Core)
      Release:            7.6.1810
      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。

```
Bash
[root@ic-admin1 ~]# batch_run -H 10.212.207.130 -c "lsb_release -a"
-o 4

>>> 10.212.207.130
      /ic/software/cad_tools/it/batchRun/tools/run_ssh_command.py -c
"ssh -XY -o StrictHostKeyChecking=no ic_admin@10.212.207.130
lsb_release \-a" -H 10.212.207.130 -p *** -t 10
      ==== output ====
      LSB Version:          :core-4.1-amd64:core-4.1-noarch:cxx-4.1-
amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-noarch:languages-
4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-
noarch
      Distributor ID:      CentOS
      Description:        CentOS Linux release 7.6.1810 (Core)
      Release:            7.6.1810
      Codename:           Core
      =====
```

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。

Bash

```
[root@ic-admin1 tools]$ ./save_password -h
usage: save_password.py [-h] [-P PASSWORD_FILE] [-u USER] -p
PASSWORD

optional arguments:
  -h, --help                show this help message and exit
  -P PASSWORD_FILE, --password_file PASSWORD_FILE
                           Specify the user password file, default is
                           "/ic/software/cad_tools/it/batch_run/config/password.encrypted".
  -u USER, --user USER    Specify user name, default is current user.
  -p PASSWORD, --password PASSWORD
                           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。

Bash

```
[root@ic-admin1 tools]$ ./get_password -h
usage: get_password.py [-h] [-P PASSWORD_FILE] [-u USER]
```

optional arguments:

```
-h, --help          show this help message and exit
-P PASSWORD_FILE, --password_file PASSWORD_FILE
                    Specify the user password file, default is
"/ic/software/cad_tools/it/batch_run/config/password.encrypted".
-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。

```
Bash
[root@ic-admin1 tools]# ./essh 10.212.206.194
spawn ssh -XY -p 22 10.212.206.194
root@10.212.206.194's password:
Last login: Sat May 21 13:28:59 2022 from 10.212.206.207
[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！

```
Bash
[root@ic-admin1 tools]# xssh 194
[FUZZY MATCH] 194 -> 10.212.206.194
xterm -e "/ic/software/cad_tools/it/batch_run/tools/essh
10.212.206.194 22" &
```

```
root@cmp194-fpga:~ (on ic-nismaster1)
spawn ssh -XY -p 22 10.212.206.194
root@10.212.206.194's password:
Last login: Thu May 26 21:12:36 2022 from 10.212.240.38
[root@cmp194-fpga ~]#
```

如果有多个匹配的 hosts，会列出供用户选择。

```
C#
[root@ic-admin1 tools]# xssh 130
[FUZZY MATCH] 130 -> 10.212.204.130
[FUZZY MATCH] 130 -> 10.212.207.130
Below are possible hosts:
  [0] 10.212.204.130
  [1] 10.212.207.130
  [2] None of above hosts

Please choice one (number): 1
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），为了更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。

```
Bash
[root@ic-admin1 batchRun]$ cat
scripts/get_server_info/get_server_info.py
#!/ic/software/tools/python3/3.8.8/bin/python3
# -*- coding: utf-8 -*-
...
import os
import re
import socket
...

```

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。

```
Bash
[root@ic-admin1 ~]$ batch_run -H 10.212.207.130 -c
get_server_info.py

>>> 10.212.207.130
    ip : 10.212.207.130
    hostname : n212-207-130
    os : CentOS Linux release 7.6.1810 (Core)
    cpu : 96
    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。

附录

附 1. 变更历史

日期	版本	变更描述	源代码变更
2022.12.12	1.0	发布第一个正式 release 版本	
2023.1.15	1.0	代码上传 https://github.com/bytedance/batchRun	更新《batchRun》用户手册
2023.1.17	1.0	支持按照 LSF 队列批量选取机器	batch_run.py: 增加新参数"--lsf_queues" common/common_lsf.py: 增加 LSF 信息解析脚本
2023.2.2	1.0	更新 LICENSE 为 GPL-v2.0	
2023.2.12	1.0	按照 PEP8 规范检查并修正 python 代码	bin/batch_run.py common/common.py common/common_lsf.py tools/encrypt_python.py tools/get_password.py tools/run_ssh_command.py tools/save_password.py tools/xssh.py