

# batchRun 用户手册

Product Name : batchRun

Product Version : V2.0

Release Date : 2024.10.26

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

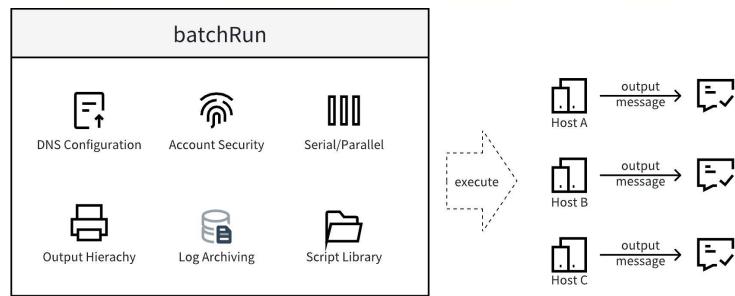
@张静文 (zhangjingwen.silvia@bytedance.com)

# 一、简介

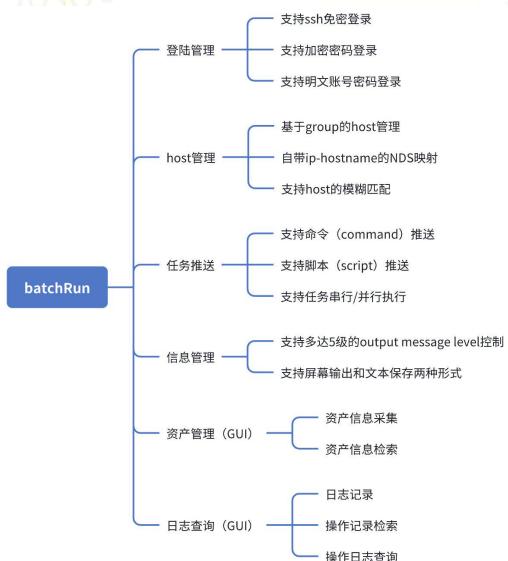
## 1.1 功能简介

batchRun 是 ByteDance 开源的一款 HPC 批量部署、资产管理和信息收集工具，采用所见即所得的命令执行方式，普遍适用于 RHEL、centos、rocky、ubuntu 等多种 Linux 操作系统环境。

batchRun 基于 python 语言开发，底层使用 ssh 实现多服务器登录，无服务端，配置和使用便捷。



batchRun 的主要功能如下：



## 1.2 快速了解

batchRun 支持 command line 模式和 GUI 模式，其中 command line 模式主要用于批量部署和信息获取，GUI 模式则额外增加了资产管理和日志查询等功能。

## 1.2.1 command line 模式

batchRun 支持命令行模式，通过参数指定的方式，在 HPC 集群中做任务的批量部署。

跟 ansible 不同的是，batchRun 采用所见即所得的运行方式，不需要借助 playbook 等配置文件，而是直接推送需要运行的 command 到远程服务器上执行，并获取返回结果。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups CMP_IC --command
hostname --output_message_level 2
>>> 10.232.132.142      n232-132-142
>>> 10.232.132.143      n232-132-143

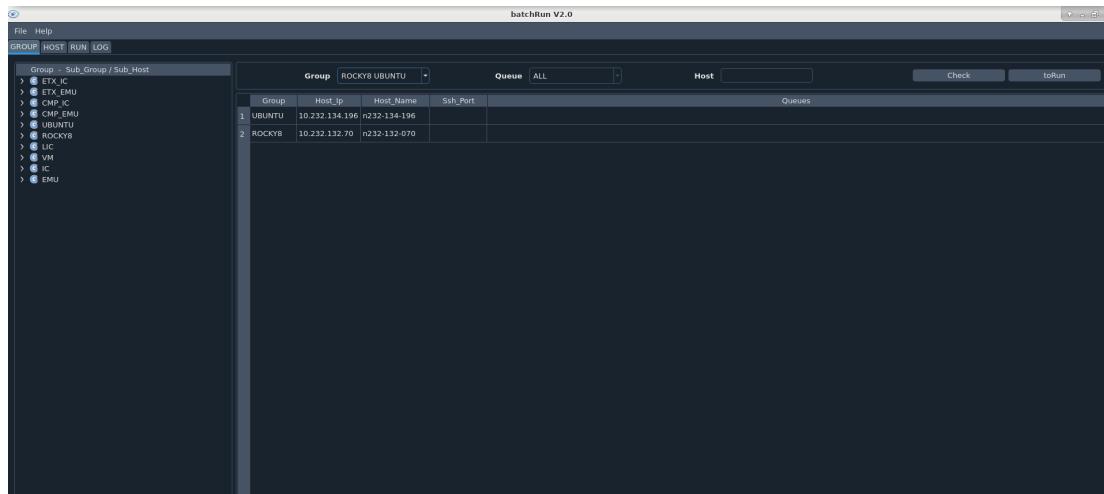
Total 2 hosts.
```

## 1.2.2 GUI 模式

batchRun 的 GUI 模式包含 GROUP/HOST/RUN/LOG 四个页面。

### GROUP 页：

用来图形化展示 host.list 的配置内容，可以看出 group <-> host <-> queue 的对应关系，可以通过 group/queue/host 三个维度来筛选 host。



## HOST 页：

用来展示所有服务器的 server\_type/os/cpu/mem 等信息，主要用作资产管理的用途。

Host_ip	Host_Name	Group	Server_Type	OS	Cpu_Arch	Cpu_Model	Cpu_Thread	Thread_Per_Core	Cpu_Freq(GHz)	MEM(GB)
1 10.232.132.146	n232-134-066	ETX_IC_IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	128	2	3.5	1007
2 10.232.132.142	n232-136-012	ETX_EMU_EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	48	1	3.1	1007
3 10.232.132.142	n232-132-142	CMP_IC_IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
4 10.232.132.143	n232-132-143	CMP_IC_IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
5 10.232.136.68	n232-136-068	CMP_EMU_EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 836C CPU @ 2.30GHz	64	1	3	2015
6 10.232.136.69	n232-136-069	CMP_EMU_EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 836C CPU @ 2.30GHz	64	1	3	2015
7 10.232.134.196	n232-134-196	UBUNTU	server	Ubuntu 20.04.6 LTS	x86_64	Intel(R) Xeon(R) Platinum 836C CPU @ 2.30GHz	128	2	3	1007
8 10.232.132.70	n232-132-070	ROCKY8	server	Rocky Linux release 8.10 (Green Obsidian)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	128	2	2.8	2015
9 10.232.158.3	i-c-k01	LIC	vm	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	4	1	2.4	7
10 10.232.158.39	i-c-monitor01	VM	vm	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	16	1	2.4	127

## RUN 页：

在选中的服务器上批量推送任务，同时可以获取命令的输出信息并展示。

The screenshot shows the 'RUN' tab of the batchRun V2.0 interface. On the left, a list of selected hosts (10.232.132.142, 10.232.132.143, 10.232.136.68, 10.232.136.69) is shown, each with a checkbox. A command input field contains 'lsb\_release -a | grep Description'. On the right, the terminal output for each host is displayed:

```
[root@iZ2t1-1 ~]# lsb_release -a
LSB Version:    :core-4.1-noarch-4.1-20240404-1
Distro:        :CentOS
Description:   :CentOS Linux release 7.9.2009 (Core)
Release:       :7.9.2009
Codename:     :Tobago
[root@iZ2t1-1 ~]#
```

At the bottom left, a log message indicates the command was run in parallel:

```
[2024-10-26 14:23:29] * Run command "lsb_release -a | grep Description" parallel with below batch run command.
[2024-10-26 14:23:29] /ic/software/tools/batchRun/bin/batch_run --hosts http://batchRun/roo/20241026142329/host.list --command "lsb_release -a | grep Description" --parallel 4 -timeout 100 --output_message_level 1 --output_file /tmp/batchRun/roo/20241026142329/HOST
[2024-10-26 14:23:41] Done
```

## LOG 页：

检索 batch\_run 的执行记录，查询每次执行的具体日志信息。

batchRun V2.0						
User		Begin_Date		End_Date		Info
Time	User	Login_User	Command	Log		
1	2024-10-26 13:04:15	root	/c/software/tools/batchRun/bin/batch_run.py -H ALL < /sb_release \a; hostnamectl; lscpu; free \g -P 30 -l 1 -o db/host_inf...	>>> 10.232.134.66 n232-134-066		
2	2024-10-26 13:16:18	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL < command /sb_release \a; hostnamectl; lscpu; free \g -parallel...	>>> n232-136-012 (10.232.136.12) n232-136-012		
3	2024-10-26 13:17:02	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL < command /sb_release \a; hostnamectl; lscpu; free \g -parallel...			
4	2024-10-26 13:20:03	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL < command /sb_release \a; hostnamectl; lscpu; free \g -parallel...			
5	2024-10-26 13:39:39	root	/c/software/tools/batchRun/bin/batch_run.py -hosts 10.232.134.66 n232.136-012 < command hostname...			
6	2024-10-26 13:41:28	root	/c/software/tools/batchRun/bin/batch_run.py -groups CMP_IC --command hostname --output_message_level 2			
7	2024-10-26 13:43:26	root	/c/software/tools/batchRun/bin/batch_run.py -groups EXT_IC CMP_IC --command hostname: touch /c/software/tools/...			
8	2024-10-26 13:44:47	root	/c/software/tools/batchRun/bin/batch_run.py -groups EXT_IC CMP_IC --command hostname: touch /c/software/tools/...			
9	2024-10-26 13:46:03	root	/c/software/tools/batchRun/bin/batch_run.py -groups EXT_IC CMP_IC --command hostname: touch /c/software/tools/...			
10	2024-10-26 13:55:39	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL < command /sb_release \a; hostnamectl; lscpu; free \g -parallel...			
11	2024-10-26 14:28:37	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026142837/host.list < command hostname...			
12	2024-10-26 14:31:04	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143103/host.list < command hostname...			
13	2024-10-26 14:31:42	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143142/host.list < command hostname...			
14	2024-10-26 14:34:04	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143404/host.list < command hostname ...			

## 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。

同时支持 RHEL/centos/rocky/ubuntu 等多种 Linux 操作系统，主要风险在于系统库版本差异可能会影响部分组件的运行。

### 2.2 python 版本依赖

batchRun 基于 python 开发，其开发和测试的 python 版本为 **python3.12.7**。

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，但过旧的 python 版本可能因为差异过大而并不支持。

### 2.3 系统组件依赖

batchRun 的部分图形界面功能依赖 xterm，xterm 如未安装则可能会造成 GUI 界面中部分组件及功能不可用。

# 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 shows the GitHub repository page for 'batchRun' owned by 'bytedance'. The repository has 18 commits, 5 forks, and 13 stars. The code tab is selected, showing a list of files and their last commit details. The 'About' section describes batchRun as an Ansible-similar IT automation system for the IC industry. It includes links to Readme, GPL-2.0 license, Activity, Custom properties, and Report repository. The 'Releases' section indicates no releases have been published. The 'Packages' section shows no packages have been published. The 'Languages' section shows Python at 99.9% and Shell at 0.1%. The bottom of the page shows the README and LICENSE files.

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

```
[bytedance@LQ2KW66YXF ~/Downloads]#git clone  
https://github.com/bytedance/batchRun.git  
Cloning into 'batchRun'...  
remote: Enumerating objects: 186, done.  
remote: Counting objects: 100% (186/186), done.  
remote: Compressing objects: 100% (119/119), done.  
remote: Total 186 (delta 65), reused 172 (delta 51), pack-reused 0  
(from 0)  
Receiving objects: 100% (186/186), 5.23 MiB | 1.16 MiB/s, done.  
Resolving deltas: 100% (65/65), done.
```

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

The screenshot shows the GitHub repository page for 'batchRun'. The 'Code' tab is selected. In the 'Clone' section, the 'GitHub CLI' tab is active. Below it, the 'Download ZIP' button is highlighted with a red box. The repository has 1 branch and 13 stars. The 'About' section describes batchRun as an Ansible-like IT automation system for the IC industry.

## 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。

batchRun 是 IT 管理员的运维自动化工具，所以建议用 root 账号将安装包拷贝到安装目录（NAS 最佳），并确保当前服务器有权限 ssh 其它被管理机器。

安装包下的文件和目录如下。

```
[root@ic-admin2 batchRun]# ls
bin  common  config  data  db  docs  install.py  LICENSE
Notice.txt  README.md  requirements.txt  scripts  tools
```

确认 python 版本正确。

```
[root@ic-admin2 batchRun]# which python3
/ic/software/tools/python3/3.12.7/bin/python3
```

使用跟 python3.12.7 对应的 pip3, 基于安装包中的 requirements.txt 安装 python 依赖库。(可能需要 root 权限)

```
[root@ic-admin2 batchRun]# which pip3
/ic/software/tools/python3/3.12.7/bin/pip3
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# pip3 install -r requirements.txt
Looking in indexes: http://bytedpypi/byted.org/simple
Requirement already satisfied: matplotlib==3.9.2 in
//ic/software/tools/python3/3.12.7/lib/python3.12/site-packages
(from -r requirements.txt (line 1)) (3.9.2)
...
```

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

```
[root@ic-admin2 batchRun]# python3 install.py
>>> Check python version.
    Required python version : (3, 12)
    Current  python version : (3, 12)

>>> Generate script "/ic/software/tools/batchRun/bin/batch_run".
>>> Generate script
"/ic/software/tools/batchRun/bin/batch_run_gui".
>>> Generate script
"/ic/software/tools/batchRun/tools/encrypt_python".
>>> Generate script "/ic/software/tools/batchRun/tools/patch".
>>> Generate script
"/ic/software/tools/batchRun/tools/sample_host_info".
>>> Generate script
"/ic/software/tools/batchRun/tools/save_password".
>>> Generate script
"/ic/software/tools/batchRun/tools/switch_etc_hosts".
>>> Generate config file
"/ic/software/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/config.py

安装目录下的 config/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。

```
# Specify host list, default is "host.list" on current configure
# directory.
host_list = '/ic/software/tools/batchRun/config/host.list'

# Specify the database directory.
db_path = '/ic/software/tools/batchRun/db'

# Default ssh command.
default_ssh_command = "ssh -o StrictHostKeyChecking=no -t -q"

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

# Define timeout for ssh command, unit is "second".
serial_timeout = 10
parallel_timeout = 100
```

**host\_list**：指定 host.list 文件位置，一般使用默认值即可。

**db\_path**：指定 batch\_run 的数据库路径。

**default\_ssh\_command**：指定默认 ssh 的命令，一般使用默认值即可。

**fuzzy\_match**：控制命令参数中 host 模糊匹配的开关，默认打开。

**serial\_timeout**：指定串行执行情况下 ssh 命令的超时时间，默认为 10 秒，如果经常执行 long-runtime 的任务，可以适当调大这个值。

**parallel\_timeout**：指定并行执行情况下 ssh 命令的超时时间，默认为 100 秒，如果经常执行 long-runtime 的任务，可以适当调大这个值。

### 3.3.2 config/host.list

host.list 用于定义所有的机器及其分组信息，下面是一个说明。

```
##### Format #####
# [group]
# host_ip1
# host_ip2 ssh_port=<ssh_port2>
# host_ip3 ssh_host=<host_name3>
# host_ip4 ssh_host=<host_name4> ssh_port=<ssh_port4>
# sub_group5
# ~host_ip6
# ~host_name7
# ~sub_group8
#####
```

host.list 配置的基本格式如下：

**[group]** #组名

**host** #机器

**sub\_group** #子组

**excluded\_host** #要排除的机器

**excluded\_sub\_group** #要排除的子组

#### 3.3.2.1 编写规则

- “group”编写规则

group 为一组机器的组名，一般用大写，不能包含空格，在 host.list 中用“[]”括起来，比如**[LOGIN]**或者**[COMPUTING]**。

- “host”编写规则

host 为机器，包含如下 4 种写法。

```
# host_ip1
# host_ip2 ssh_port=<ssh_port2>
# host_ip3 ssh_host=<host_name3>
# host_ip4 ssh_host=<host_name4> ssh_port=<ssh_port4>
```

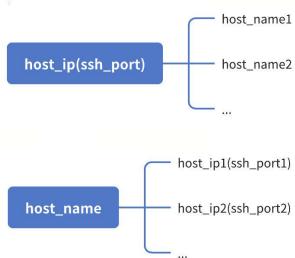
`host_ip` 是必选项，是 `host` 的确定标识。

`host_name` 是可选项，主要用于标识 DNS 映射关系。

`ssh_port` 是可选项，如果 `host_ip` 的 ssh 端口不是 22，则需要在此处指定。

此处 `host_name` 需要注意，不能包含空格。

为保持和/etc/hosts 的兼容，`host_ip` 和 `host_name` 是多对多的对应关系，即一个 `host_ip` 可能对应多个多个 `host_name`（但是对应唯一的 `ssh_port`），一个 `host_name` 也可能对应多个 `host_ip`，对应关系如下。



- “`sub_group`”编写规则

`sub_group` 为多机器的组名，不能包含空格，必须在 `host.list` 文件中被定义过。

- “`excluded_host`”编写规则

可以按照 `host_ip` 或者 `host_name` 来排除 host，在前面加上波浪线“~”即可，样式如下。

```
[group]
~host_ip6
~host_name7
```

`~host_ip2`: [group]中把所有的项按照 `host_ip` 的维度展开，如果存在 `host_ip2`，则会被排除。

`~host_name3`: [group]中把所有的项按照 `host_ip` 的维度展开，如果某个 `host_ip` 的机器名为 `host_name3`，则会被排除。

- “excluded\_sub\_group”编写规则

可以按照 sub\_group 来排除 host, 在前面加上波浪线“~”即可, 样式如下。

```
[group]
~sub_group8
```

~sub\_group4: [group]中把所有的项按照 host\_ip 的维度展开, sub\_group4 中的 host\_ip 会被排除掉。

### 3.3.2.2 列表生成

config/host.list 可以手工配置, 但是在服务器数量较多时较为繁琐, 所以更推荐基于格式化的/etc/hosts 文件直接生成, 转换方法参照#5.1 tools/switch\_etc\_hosts 章节。

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts -i /etc/hosts -o
config/host.list

Output File : config/host.list
[root@ic-admin2 batchRun]# cat config/host.list

[LOGIN_IC]
10.232.134.66 ssh_host=n232-134-066
...
```

### 3.3.3 config/password.encrypted

这个文件由自带工具 tools/save\_password 生成, 用户可以为自己配置加密密码, 详情可以参照#5.2 章节。

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。

```
[root@ic-admin2 batchRun]# tools/save_password -p ***
[root@ic-admin2 batchRun]# cat db/password/root
root default 1724049499 ed5fa5a1d09c8f4bf3e99c55708be23a
00b2a6273de7bc43a83f9bc352822827
```

## 3.4 服务器信息采样

batchRun 的资产管理功能，依赖于 sample\_host\_info 的服务器信息采样，采样方法可以参照#5.3 tools/sample\_host\_info 章节。

```
[root@ic-admin2 batchRun]# tools/sample_host_info --hosts ALL --parallel 20 --output_dir db/host_info  
>>> Move /ic/software/tools/batchRun/db/host_info/host_info.json  
to  
/ic/software/tools/batchRun/db/host_info/host_info.json.20241026  
_132002 ...  
mv /ic/software/tools/batchRun/db/host_info/host_info.json  
/ic/software/tools/batchRun/db/host_info/host_info.json.20241026  
_132002  
>>> Clean up /ic/software/tools/batchRun/db/host_info ...  
rm -rf *.info host_list.json host_info.json  
>>> Sampling host list information ...  
/ic/software/tools/batchRun/bin/batch_run --hosts ALL --list  
--output_file  
/ic/software/tools/batchRun/db/host_info/host_list.json  
* Host(s) info has been saved into  
"/ic/software/tools/batchRun/db/host_info/host_list.json".  
>>> Sampling host os/cpu/mem information ...  

```

如果资产信息经常处于变动之中，建议用 crontab 设置一个定时采样，以确保资产信息可以被及时更新。

```
# For batchRun, sample host information.  
10 0 * * * /ic/software/tools/batchRun/tools/sample_host_info --  
groups ALL --parallel 1000 --output_dir  
/ic/software/tools/batchRun/db/host_info
```

## 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。

```
[root@ic-admin2 batchRun]# module load cad
[root@ic-admin2 batchRun]# which batch_run
/ic/software/tools/batchRun/bin/batch_run
```

### 4.2 command line 功能介绍

#### 4.2.1 帮助信息

batch\_run 的帮助信息如下。

```
[root@ic-admin2 batchRun]# bin/batch_run -h
usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-L] [-u USER] [-p PASSWORD] [-c COMMAND
[COMMAND ...]] [-s SCRIPT [SCRIPT ...]] [-P PARALLEL] [-t TIMEOUT]
[-l {0,1,2,3,4}] [-o OUTPUT_FILE] [-g] [-v]

options:
-h, --help            show this help message and exit
-H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
Specify host(s) with below format:
<host_ip_file>
<host_ip>
<host_ip>:<ssh_port>
<host_name>
<host_name>:<ssh_port>
~<host_ip>
~<host_name>

"all | ALL" means all hosts on
/ic/software/tools/batchRun/config/host.list.
"~<host>" means exclud specified host.
-G GROUPS [GROUPS ...], --groups GROUPS [GROUPS ...]
```

```
Specify host group(s) with below format:  
<group_file>  
<group>  
~<group>  
  
"all | ALL" means all groups on  
/ic/software/tools/batchRun/config/host.list.  
"~<GROUP>" means exclud specified group.  
-L, --list List specified host(s)/group(s).  
-u USER, --user USER Specify the user identity for SSH login to  
specified host.  
-p PASSWORD, --password PASSWORD  
Specify the user password for SSH login to  
specified host.  
-c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]  
Specify the command to run on specified  
remote host(s).  
-s SCRIPT [SCRIPT ...], --script SCRIPT [SCRIPT ...]  
Specify the script to run on specified  
host(s), will copy script to host /tmp directory.  
-P PARALLEL, --parallel PARALLEL  
Specify the parallelism of command  
execution with a number.  
-t TIMEOUT, --timeout TIMEOUT  
Specify the timeout for SSH, which  
defaults to 10 seconds in serial and 100 seconds in parallel.  
-l {0,1,2,3,4}, --output_message_level {0,1,2,3,4}  
Specify output message level, which  
defaults to "3" in serial and "0" in parallel.  
"0" : print host info;  
"1" : print command output message;  
"2" : print host info and the first line  
of the command output message;  
"3" : print host info and complete command  
output message;  
"4" : print verbose information with ssh  
command.  
-o OUTPUT_FILE, --output_file OUTPUT_FILE  
Export output message of command/script to  
specified file instead of on the screen.  
-g, --gui Open batchRun with GUI format.  
-v, --version Show batchRun version information.
```

**--help** : 打印帮助信息。

**--hosts** : 指定机器列表，可以是机器 ip，可以是机器 hostname，也可以是一个文件中包含了 host\_ip/host\_name。如果是“ALL”，则意味着配置文件 host.list 中所有的机器。

**--groups** : 根据服务器组获取服务器列表，也可以是一个文件中包含了服务器组信息。如果是“ALL”，则意味着配置文件 host.list 中所有的机器。

**--list**: 列出指定的 hosts 或者 groups。

**--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。

**--script** : 到远程机器上要执行的脚本，执行方式跟 command 一样，但是脚本本身会被拷贝到远程机器上的 /tmp 目录下再执行。

**--parallel** : 指定并行运行模式的并行数量，如未指定则为串行运行模式。

**--timeout** : 指定 ssh 的超时时间，串行模式默认为 10 秒，并行模式默认为 100 秒。

**--output\_message\_level** : 指定执行命令时输出信息的详细程度，分为 0-4 共五个等级。

“0”：只打印机器名。

“1”：只打印（执行命令的）输出信息。

“2”：打印机器名和（执行命令的）输出信息的第一行。

“3”：打印机器名和完成的（执行命令的）输出信息。

“4”：在#3 的基础上打印更详细的 debug 信息（ssh 命令）。

**--output\_file** : 指定输出文件，如指定，会将输出在屏幕上的（执行命令的）输出信息导入到指定的输出文件中。

**--gui** : 启动 GUI 界面。

**--version** : 打印 batchRun 版本信息。

## 4.2.2 打印 batchRun 版本信息。

batch\_run --version 可以打印 batchRun 的版本信息，当前版本是 V1.1，是 V1.0 的小升级版本。

```
[root@ic-admin2 batchRun]# bin/batch_run --version
Version : V2.0
Release : 2024.10.26
```

## 4.2.3 列出指定机器

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。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --hosts ALL
GROUP : [ETX_IC]
        10.232.134.66      n232-134-066
GROUP : [ETX_EMU]
        10.232.136.12      n232-136-012
GROUP : [CMP_IC]
        10.232.132.142     n232-132-142
        10.232.132.143     n232-132-143
GROUP : [CMP_EMU]
        10.232.136.68      n232-136-068
        10.232.136.69      n232-136-069
GROUP : [UBUNTU]
        10.232.134.196     n232-134-196
GROUP : [ROCKY8]
        10.232.132.70      n232-132-070
GROUP : [LIC]
        10.232.158.3       ic-lic01
GROUP : [VM]
        10.232.158.39      ic-monitor01  8888
```

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

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC
GROUP : [CMP_IC]
        10.232.132.142     n232-132-142
```

#### 4.2.4 指定用户名和密码登录远程机器

如果知道用户名和密码，可以采用指定用户名和密码的方式，以指定帐号的身份登录远程机器执行命令，操作方式为“**batch\_run --user <USER> --password <PASSWORD> ...**”。

下面是一个示例，root 用指定账号密码登录机器并执行命令。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
user liyanqing.1987 --password *** --command whoami

>>> 10.232.134.66
liyanqing.1987
```

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。

#### 4.2.5 采用内置加密密码登录远程机器

使用内置工具 `save_password` 为指定账号保存加密密码。

```
[liyanqing.1987@n232-135-013 batchRun]$ tools/save_password --
password ***
```

其储存的密码为加密样式。

```
[liyanqing.1987@n232-135-013 batchRun]$ cat
db/password/liyanqing.1987
liyanqing.1987 default 1724392876
a2c768667257c39f4bc2a1a4fa1124dc 8d6f948e98fa7ba36cfa07f5d72afafa
```

此时，`batch_run` 用当前 Linux 登录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。

```
[liyanqing.1987@n232-135-013 batchRun]$ cat  
db/password/liyanqing.1987  
liyanqing.1987 default 1724392876  
a2c768667257c39f4bc2a1a4fa1124dc 8d6f948e98fa7ba36cfa07f5d72afafa
```

请注意，上面例子中，一定要以密码登陆的方式成为 liyanqing.1987 用户，否则 batch\_run 会无法判断当前实际用户是否知悉 liyanqing.1987 账号密码（比如 root 账号 su 成 liyanqing.1987，并不需要知道 liyanqing.1987 的密码），从而导致拒绝执行命令。

此时即便是没有配置 ssh 免密，也不输入明文密码，仍然可以方便地 ssh 地通过 batchRun 登录到远程服务器上执行任务。

```
[liyanqing.1987@n232-135-013 batchRun]$ bin/batch_run --hosts  
10.232.134.66 --command whoami  
  
>>> 10.232.134.66  
liyanqing.1987
```

## 4.2.6 指定机器执行命令

batch\_run 支持用“batch\_run --hosts <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）。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66  
n232-136-012 --command hostname  
  
>>> 10.232.134.66  
n232-134-066  
  
>>> n232-136-012 (10.232.136.12)  
n232-136-012  
  
Total 2 hosts.
```

同时，batch\_run 的--hosts 参数也支持模糊匹配，会把所有匹配的项都列出来。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --hosts 132
[FUZZY MATCH] 132 -> 10.232.132.142
[FUZZY MATCH] 132 -> 10.232.132.143
[FUZZY MATCH] 132 -> 10.232.132.70

GROUP : [CMP_IC]
      10.232.132.142    n232-132-142
      10.232.132.143    n232-132-143
GROUP : [ROCKY8]
      10.232.132.70     n232-132-070
```

不但可以匹配 host ip, 也可以匹配 host name, 还支持^\$等正则匹配符号来界定位置。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --hosts ^n232-134
[FUZZY MATCH] ^n232-134 -> n232-134-066 -> 10.232.134.66
[FUZZY MATCH] ^n232-134 -> n232-134-196 -> 10.232.134.196

GROUP : [ETX_IC]
      10.232.134.66    n232-134-066
GROUP : [UBUNTU]
      10.232.134.196   n232-134-196
```

#### 4.2.7 指定机器组执行命令

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

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC
CMP_EMU
GROUP : [CMP_IC]
      10.232.132.142    n232-132-142
      10.232.132.143    n232-132-143
GROUP : [CMP_EMU]
      10.232.136.68     n232-136-068
      10.232.136.69     n232-136-069
```

也可以采用除外的方式排除掉指定的 group 或者 host。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC  
CMP_EMU --hosts ~10.232.132.143 ~n232-136-069  
GROUP : [CMP_IC]  
        10.232.132.142    n232-132-142  
GROUP : [CMP_EMU]  
        10.232.136.68    n232-136-068
```

## 4.2.8 执行命令 (command)

batch\_run 可以在远程机器上执行命令，此处的命令既可以表示系统命令（比如 hostname），也可以表示脚本命令（比如~/.run.sh），前提是远程机器上命令是可以直接访问和执行的。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups CMP_IC --command  
hostname --output_message_level 2  
>>> 10.232.132.142      n232-132-142  
>>> 10.232.132.143      n232-132-143  
  
Total 2 hosts.
```

## 4.2.9 执行脚本 (script)

script 跟 command 的主要区别在于，我们认为 command 在远程机器上是一定存在且可执行的，但是 script 则未必（比如 script 是本地脚本），因此需要拷贝到远程机器上再执行。

下面是一个本地脚本，只在本机存在。

```
[root@ic-admin2 batchRun]# cat ~/local_run.sh  
#!/bin/bash  
  
echo "I am `whoami`"  
echo "This host is `hostname`"
```

如果想在远程机器上执行 local\_run.sh，只能将它拷贝到远程机器上再执行。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.132.142 --  
script ~/local_run.sh --output_message_level 4
```

```
>>> 10.232.132.142
    scp /root/local_run.sh 10.232.132.142:/tmp/local_run.sh
    ssh -o StrictHostKeyChecking=no -t -q root@10.232.132.142
/tmp/local_run.sh
===== output =====
I am root
This host is n232-132-142
=====
Total 1 hosts.
```

#### 4.2.10 串行执行和并行执行

如未指定--parallel 参数，batch\_run 的任务默认是串行执行。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups ETX_IC CMP_IC --
command 'hostname; touch
/ic/software/tools/batchRun/db/test/`hostname`' --
output_message_level 2
>>> 10.232.134.66      n232-134-066
>>> 10.232.132.142      n232-132-142
>>> 10.232.132.143      n232-132-143
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# ls /ic/software/tools/batchRun/db/test
n232-132-142  n232-132-143  n232-134-066
```

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，添加--parallel 参数用并行方式再执行一遍。（此时 output\_message\_level 被强制为“0”）

```
[root@ic-admin2 batchRun]# rm -rf
/ic/software/tools/batchRun/db/test/*
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# bin/batch_run --groups ETX_IC CMP_IC --
command 'hostname; touch
/ic/software/tools/batchRun/db/test/`hostname`' --parallel 3
*Warning*: Switch output_message_level to "0" on parallel mode.
>>> 10.232.134.66
```

```
>>> 10.232.132.142
>>> 10.232.132.143

Total 3 hosts.
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# ls /ic/software/tools/batchRun/db/test
n232-132-142  n232-132-143  n232-134-066
```

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。

#### 4.2.11 超时时间 (timeout)

batch\_run 底层基于 ssh 登陆远程服务器执行命令，串行运行时其默认的 timeout 为 10 秒，并行运行时其默认的 timeout 为 100 秒，如果任务的执行时间超过 timeout 时间，ssh 会报告超时并强制断开连接。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups ETX_IC --command
"sleep 15"

>>> 10.232.134.66
    Timeout exceeded.
    ...
0: re.compile(b'assword:')
1: EOF

Total 1 hosts.
```

执行运行时间超过 timeout 的任务时，可以通过设置更长的 timeout 时间来规避这个问题。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups ETX_IC --command
"sleep 15" --timeout 20

>>> 10.232.134.66

Total 1 hosts.
```

## 4.2.12 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，仅打印服务器名。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --  
command "free -g" --output_message_level 0  
>>> 10.232.134.66  
  
Total 1 hosts.
```

- “1”级输出，仅打印执行命令的输出信息，主要是配置--output\_file 参数使用。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --  
command "free -g" --output_message_level 1  
total          used          free          shared    buff/cache  
available  
Mem:        1007           58           16            0  
931         946  
Swap:        127            6           120  
  
Total 1 hosts.
```

- “2”级输出，打印服务器名，并显示执行命令的输出信息的第一行。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --  
command "free -g" --output_message_level 2  
>>> 10.232.134.66      total          used          free          shared  
buff/cache   available  
  
Total 1 hosts.
```

- “3”级输出，默认级别，打印服务器名，并显示执行命令的输出信息的全部。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --  
command "free -g" --output_message_level 3  
>>> 10.232.134.66  
total          used          free          shared    buff/cache
```

```
available
    Mem:          1007           58           16           0
931      946
    Swap:          127            6          120

Total 1 hosts.
```

- “4”级输出，在#3 的基础上，会打印实际执行的 ssh 命令，以方便用户手工执行并复现结果。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 4

>>> 10.232.134.66
    ssh -o StrictHostKeyChecking=no -t -q root@10.232.134.66 free
\ -g
    ===== output =====
    total        used         free        shared   buff/cache
available
    Mem:          1007           58           16           0
931      946
    Swap:          127            6          120
=====
Total 1 hosts.
```

#### 4.2.13 输出文件

batch\_run 的任务执行结果默认输出到屏幕上，但是也可以通过--output\_file 参数将其导入指定文件中。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 1 --output_file
10.232.134.66.mem

Total 1 hosts.
[root@ic-admin2 batchRun]# cat 10.232.134.66.mem
    total        used         free        shared   buff/cache
available
    Mem:          1007           58           16           0
```

931	946
Swap:	127
	6
	120

batch\_run 还支持自动将--output\_file 中的“HOST”字符自动转换为实际的 host 名，这样非常利于多服务器信息采样的信息保存。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups ETX_IC CMP_IC --  
command "free -g" --output_message_level 1 --output_file  
db/host_info/HOST.mem  
  
Total 3 hosts.  
[root@ic-admin2 batchRun]# ls db/host_info  
10.232.132.142.mem 10.232.132.143.mem 10.232.134.66.mem  
[root@ic-admin2 batchRun]# cat db/host_info/10.232.134.66.mem  
total used free shared buff/cache  
available  
Mem:  
931 946  
Swap:  
127 6 120
```

### 4.3 GUI 功能介绍

执行“batch\_run --gui”可以开启其 GUI 界面，GUI 界面查看信息和执行命令相对而言更直观。

```
[root@ic-admin2 batchRun]# bin/batch_run --gui  
[2024-10-26 13:55:48] Getting LSF host-queue relationship, please  
wait a moment ...
```

	Host_Id	Host_ip	Host_Name	Group	Server_Type	OS	Cpu_Arch	Cpu_Model	Cpu_Thread	Thread_Per_Core	Cpu_Freq(GHz)	MEM(GB)
1	10.232.134.66	n232-134-066	ETX_IC	IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	128	2	3.5	1007
2	10.232.136.12	n232-136-012	ETX_EMU	EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	48	1	3.1	1007
3	10.232.132.142	n232-132-142	CMP_IC	IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
4	10.232.132.143	n232-132-143	CMP_IC	IC	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
5	10.232.136.68	n232-136-068	CMP_EMU	EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8360C CPU @ 2.30GHz	64	1	3	2015
6	10.232.136.69	n232-136-069	CMP_EMU	EMU	server	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8360C CPU @ 2.30GHz	64	1	3	2015
7	10.232.134.196	n232-134-196	UBUNTU		server	Ubuntu 20.04.4 LTS	x86_64	Intel(R) Xeon(R) Platinum 8360C CPU @ 2.30GHz	128	2	3	1007
8	10.232.132.70	n232-132-070	ROCKY8		server	Rocky Linux release 8.10 (Green Obsidian)	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	128	2	2.8	2015
9	10.232.158.3	ic-lc01	LIC		vm	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	4	1	2.4	7
10	10.232.158.39	ic-monitor01	VM		vm	CentOS Linux release 7.9.2009 (Core)	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	16	1	2.4	127

batchRun 的 GUI 界面共有包含 GROUP/HOST/RUN/LOG 四个界面，下面一一介绍。

### 4.3.1 GROUP 页

GROUP 页用来展示 group <-> host <-> queue 的关系。

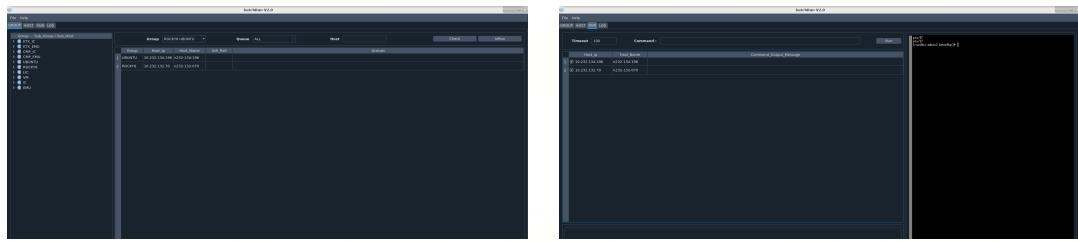
左侧边栏是 host.list 配置的图形化重现，体现了 group 的定义关系，即每个 group 包含了哪些子 group 和子 host。

右侧的表格则按照 group 的维度来展示 group -> host 的归属关系，同时，如果 host 属于某个 LSF queue，则把相应的 queue 信息也展示出来。

Group	Sub_Group/Sub_Host	Group	Host_ip	Host_Name	Ssh_Port	Queue
> ETX_IC		1	10.232.134.66	n232-134-066		
> ETX_EMU		2	10.232.136.12	n232-136-012		
> CMP_IC		3	10.232.132.142	n232-132-142		
> CMP_EMU		4	10.232.132.143	n232-132-143		
> UBUNTU		5	10.232.136.68	n232-136-068		
> ROCKY8		6	10.232.136.69	n232-136-069		
> LIC		7	10.232.134.196	n232-134-196		
> VM		8	10.232.132.70	n232-132-070		
> IC		9	10.232.158.3	ic-lc01		
> EMU		10	10.232.158.39	ic-monitor01	8888	
		11	10.232.134.66	n232-134-066		
		12	10.232.132.142	n232-132-142		
		13	10.232.132.143	n232-132-143		
		14	10.232.136.12	n232-136-012		
		15	10.232.136.68	n232-136-068		
		16	10.232.136.69	n232-136-069		

右侧的表格可以按照 group/queue/host 来做筛选。

筛选过后的 host(s)，点击“toRun”按钮，可以将它们导入到 RUN 页面。



### 4.3.2 HOST 页

HOST 页用来展示 host 的基本信息，包括服务器类型、操作系统、cpu/mem 等，可以起到基本的资产管理的作用。

batchitun V2.0										
File Help		batchitun V2.0								
GROUP		HOST		RUN		LOG				
Host_Id	Host_Name	Group	Server_Type	OS	Cpu_Arch	Cpu_Model	Cpu_Threads	Thread_Per_Core	Cpu_Freq(GHz)	Mem(GB)
Cpu_Thread	ALL	ALL	ALL	ALL	Cpu_Freq	ALL	ALL	ALL	ALL	ALL
ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
1	10.232.134.66	n232-134-066	ETX_IC_IC	server	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	128	2	3.5	1007
2	10.232.136.12	n232-136-012	ETX_EMU_EMU	server	x86_64	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz	48	1	3.1	1007
3	10.232.132.142	n232-132-142	CMP_IC_IC	server	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
4	10.232.132.143	n232-132-143	CMP_IC_IC	server	x86_64	Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80GHz	64	1	3.5	2015
5	10.232.136.68	n232-136-068	CMP_EMU_EMU	server	x86_64	Intel(R) Xeon(R) Platinum 836C CPU @ 2.30GHz	64	1	3	2015
6	10.232.136.69	n232-136-069	CMP_EMU_EMU	server	x86_64	Intel(R) Xeon(R) Platinum 836C CPU @ 2.30GHz	64	1	3	2015
7	10.232.134.196	n232-134-196	UBUNTU	server	x86_64	Ubuntu 20.04.4 LTS	128	2	3	1007
8	10.232.132.70	n232-132-070	ROCKY8	server	x86_64	Rocky Linux release 8.10 (Green Obsidian)	128	2	2.8	2015
9	10.232.158.3	i-clic01	UC	vm	x86_64	CentOS Linux release 7.9.2009 (Core)	4	1	2.4	7
10	10.232.158.39	ic-monitor01	VM	vm	x86_64	CentOS Linux release 7.9.2009 (Core)	16	1	2.4	127

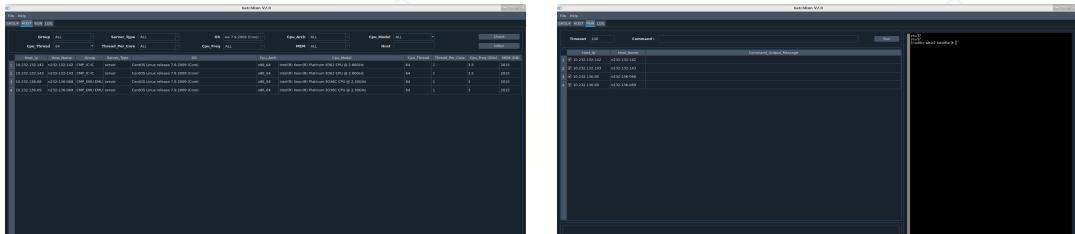
下侧表格中展示的信息具体如下。

列标题	信息
Host_Ip	服务器 ip
Host_Name	服务器 hostname
Group	服务器在 host.list 中的归属组
Server_Type	服务器类型，一般分为 server 和 vm 两种
OS	操作系统

Cpu_Arch	cpu 架构
Cpu_Model	cpu 型号
Cpu_Thread	cpu 核数
Thread_Per_Core	cpu core 的线程数，判断超线程是否开启
Cpu_Freq (GHz)	cpu 频率，单位为 GHz
MEM (GB)	内存容量

上侧的下拉菜单中，如果包含为空的可选项，说明部分服务器采样失败，采集到的信息为空。

表格内容可以按照 group/server\_type/os 等多个维度来筛选，筛选过后的 host(s)，点击“toRun”按钮，可以将它们导入到 RUN 页面。



### 4.3.3 RUN 页

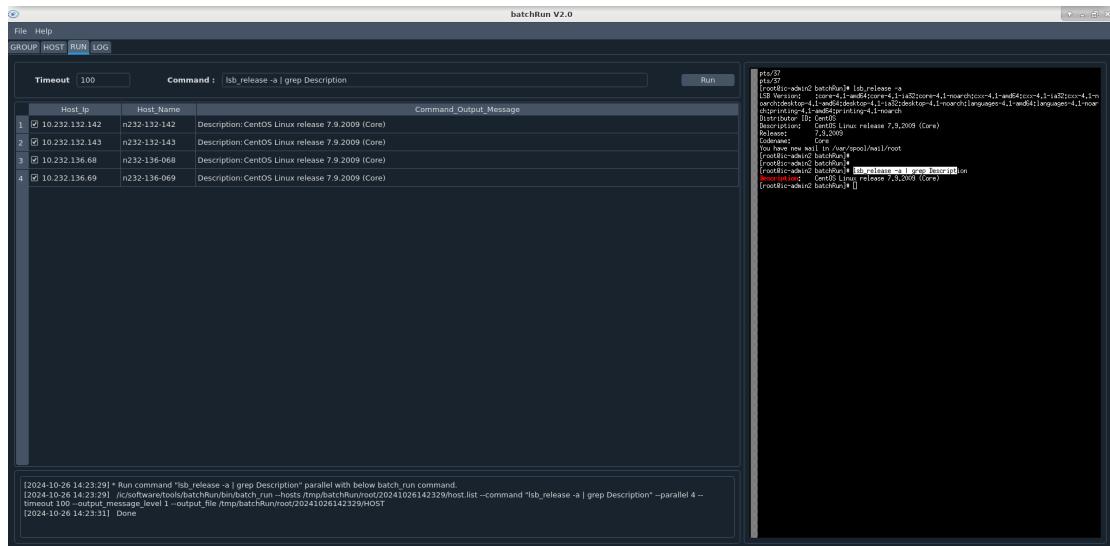
RUN 页用来在选中的服务器上批量运行同一命令，同时可以直观地获取命令执行后的返回信息。

右侧为嵌入的 xterm 工具，可以将执行命令在本地做测试，以验证期望效果。

上侧可以设定 ssh 的 timeout，以及要执行的 command，点击“Run”按钮开始并行执行。

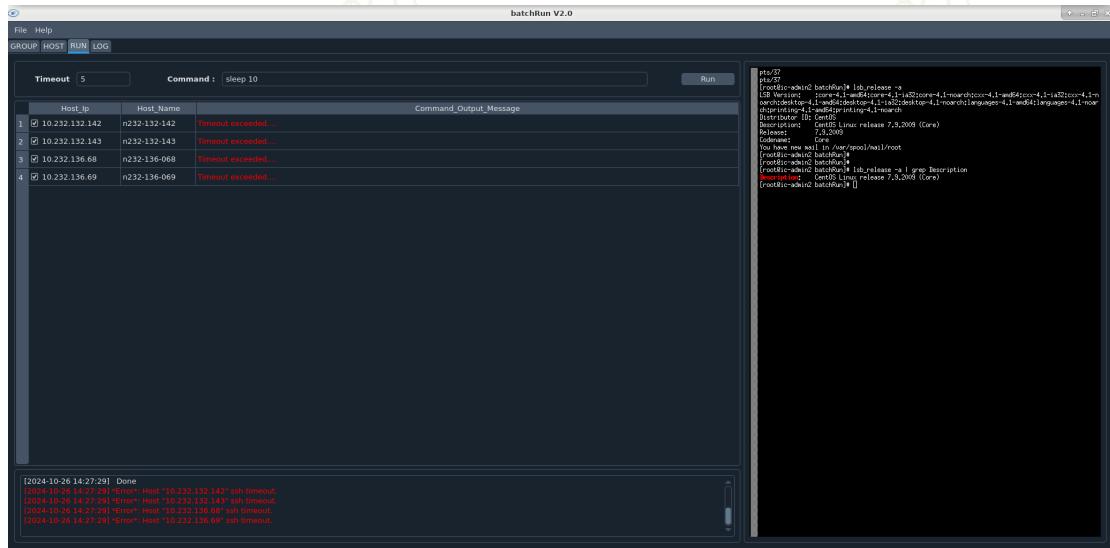
执行完毕后，每台机器上 command 的 output message 会输出在表格中，可以起到信息收集的作用。

操作的过程会显示在下侧的日志框中。

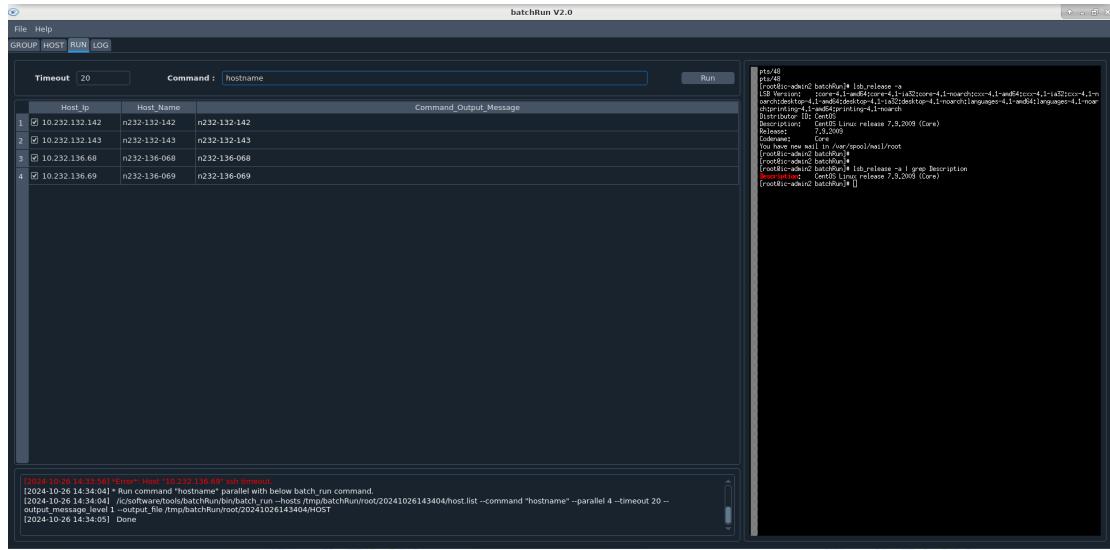


中间的表格，点击“Host\_Ip”可以全选/全部取消下面的服务器选项，每台服务器也可以单独设定选中状态，仅选中的服务器上会执行指定的 command。

下方的日志框，针对 ssh 失败，或者 hostname 不匹配等异常情况，会以 Warning/Error 信息的形式提示，比如人为制造一种服务器登录执行超时的情况，日志框会提示 timeout 的问题。

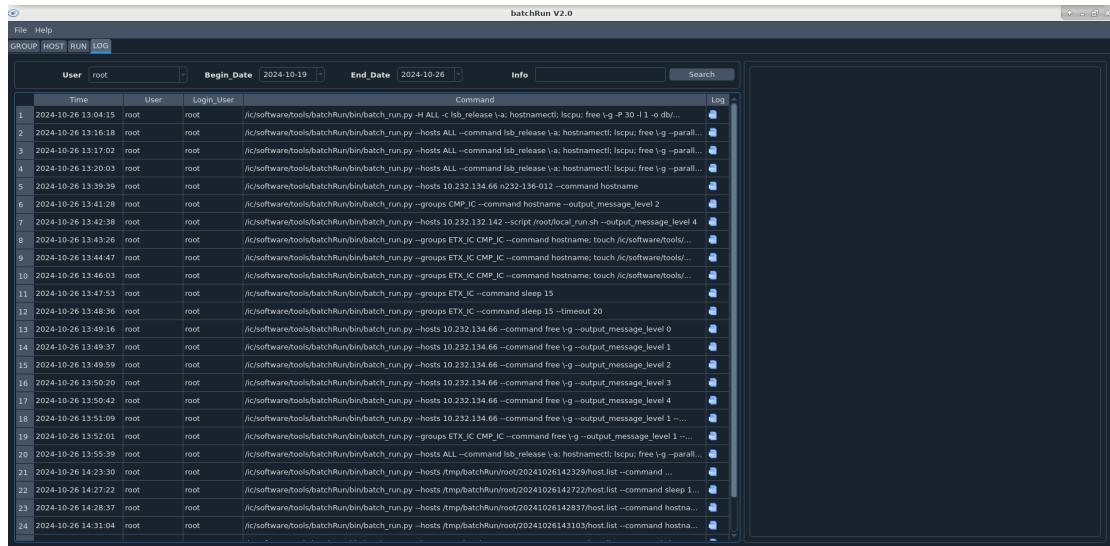


这一功能也可以用来验证/etc/hosts (host.list) 中的 host\_ip 和 host\_name 对应关系是否正确。



#### 4.3.4 LOG 页

LOG 页用来检索和查询 batchRun 的操作记录，其中 root 账号可以查看所有人的操作记录，普通账号则仅可以查看自己的操作记录。



如果只对指定的 command 感兴趣，可以通过“Info”项做一次过滤，点击每一条记录最后侧的 Log 图标，则可以在右侧的文本框中展示当次操作的 log 内容。

batchRun V2.0

File Help RUN LOG

GROUP HOST RUN LOG

User	root	Begin Date	2024-10-19	End Date	2024-10-26	Info	hostname	Search
Time	User	Login_User	Command	Log				
1	2024-10-26 13:04:15	root	root	/c/software/tools/batchRun/bin/batch_run.py -H ALL -c lib_release \a; hostnamecl; lscpu; free \g -P 30 -l 1 -o db/host_info				
2	2024-10-26 13:16:18	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL -command lib_release \a; hostnamecl; lscpu; free \g -parallel ...				
3	2024-10-26 13:17:02	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL -command lib_release \a; hostnamecl; lscpu; free \g -parallel ...				
4	2024-10-26 13:20:03	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL -command lib_release \a; hostnamecl; lscpu; free \g -parallel ...				
5	2024-10-26 13:39:39	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts 10.232.134.66 n232.136.012 -command hostname				
6	2024-10-26 13:41:28	root	root	/c/software/tools/batchRun/bin/batch_run.py -groups ETX_IC CMP_IC -command hostname -output_message_level 2				
7	2024-10-26 13:45:26	root	root	/c/software/tools/batchRun/bin/batch_run.py -groups ETX_IC CMP_IC -command hostname; touch /c/software/tools/...				
8	2024-10-26 13:44:47	root	root	/c/software/tools/batchRun/bin/batch_run.py -groups ETX_IC CMP_IC -command hostname; touch /c/software/tools/...				
9	2024-10-26 13:46:03	root	root	/c/software/tools/batchRun/bin/batch_run.py -groups ETX_IC CMP_IC -command hostname; touch /c/software/tools/...				
10	2024-10-26 13:55:39	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts ALL -command lib_release \a; hostnamecl; lscpu; free \g -parallel ...				
11	2024-10-26 14:28:37	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026142837/host.list -command hostname				
12	2024-10-26 14:31:04	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143103/host.list -command hostname				
13	2024-10-26 14:31:42	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143142/host.list -command hostname				
14	2024-10-26 14:34:04	root	root	/c/software/tools/batchRun/bin/batch_run.py -hosts /tmp/batchRun/root/20241026143404/host.list -command hostname				

## 五、辅助工具

batchRun 的辅助工具位于 tools 目录下。

### 5.1 tools/switch\_etc\_hosts

工具 switch\_etc\_hosts 用于将格式化的/etc/hosts 文件转换为 batchRun 的 host.list 文件，用法如下。

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts -h
usage: switch_etc_hosts.py [-h] [-e EXPECTED_IPS
[EXPECTED_IPS ...]] [-E EXCLUDED_IPS [EXCLUDED_IPS ...]] [-i
INPUT_FILE] [-o OUTPUT_FILE] [-a APPEND] [-r] [-t
{batchRun,ansible}]

optional arguments:
  -h, --help            show this help message and exit
  -e EXPECTED_IPS [EXPECTED_IPS ...], --expected_ips EXPECTED_IPS
[EXPECTED_IPS ...]
                                         Specify expected ip(s), support regular
                                         expressions.
  -E EXCLUDED_IPS [EXCLUDED_IPS ...], --excluded_ips EXCLUDED_IPS
[EXCLUDED_IPS ...]
                                         Specify excluded ip(s), support regular
                                         expressions.
  -i INPUT_FILE, --input_file INPUT_FILE
                                         Specify input file, default is
                                         "/etc/hosts".
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                                         Specify output file, default is
                                         "./host.list".
  -a APPEND, --append APPEND
                                         Append configuration file into output
                                         file.
  -r, --rewrite          Rewrite mode, rewrite output file by
force.
  -t {batchRun,ansible}, --tool {batchRun,ansible}
                                         Which tool the host list is for, default
                                         is "batchRun".
```

**--expected\_ips:** 指定期望的 ip，也可以用正则匹配的方式指定 ip 段 (10.212.206.

\*)，如指定，那么 hosts 文件中不在指定列表中 ip 会被抛弃。

**--excluded\_ips**: 指定排除的 ip，如指定，那么 hosts 文件中在指定列表中的 ip 会被抛弃。

**--input\_file**: 指定输入 hosts 文件来源，默认是“/etc/hosts”。

**--output\_file**: 指定输出 host.list 文件路径，默认是“./host.list”。

**--append**: 指定追加文件，会以文本的形式拼接在 output\_file 结尾，一般用于指定组嵌套关系。

**--rewrite**: 强制覆盖模式，如指定，output\_file 在已存在的情况下会被强制覆盖。

**--tool**: 指定输出 host.list 的格式，默认是 batchRun 所支持的格式，也可以选 ansible 所支持的格式。

/etc/hosts 文件需要符合如下格式才能被正确解析。

```
# GROUP : <group>
<host_ip> <host_name>
<host_ip> <host_name> # SSH_PORT=<port>
```

下面是一个示例 hosts 文件。

```
[root@ic-admin2 batchRun]# cat config/hosts
# GROUP : ETX_IC
10.232.134.66 n232-134-066

# GROUP : ETX_EMU
10.232.136.12 n232-136-012

# GROUP : CMP_IC
10.232.132.142 n232-132-142
10.232.132.143 n232-132-143

# GROUP : CMP_EMU
10.232.136.68 n232-136-068
10.232.136.69 n232-136-069

# GROUP : UBUNTU
10.232.134.196 n232-134-196

# GROUP : ROCKY8
```

```
10.232.132.70    n232-132-070

# GROUP : LIC
10.232.158.3   ic-lic01

# GROUP : VM
10.232.158.39  ic-monitor01 # SSH_PORT=8888
```

我们将如上 hosts 文件转换为 batchRun 的 host.list 文件。(也支持转换为 ansible 格式的 hosts 文件)

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts --input_file
config/hosts --output_file config/host.list --rewrite --tool
batchRun

Output File : config/host.list
```

转换后的 batchRun host.list 文件如下。

```
[ETX_IC]
10.232.134.66  ssh_host=n232-134-066

[ETX_EMU]
10.232.136.12  ssh_host=n232-136-012

[CMP_IC]
10.232.132.142  ssh_host=n232-132-142
10.232.132.143  ssh_host=n232-132-143

[CMP_EMU]
10.232.136.68  ssh_host=n232-136-068
10.232.136.69  ssh_host=n232-136-069

[UBUNTU]
10.232.134.196  ssh_host=n232-134-196

[ROCKY8]
10.232.132.70  ssh_host=n232-132-070

[LIC]
10.232.158.3   ssh_host=ic-lic01
```

```
[VM]
10.232.158.39 ssh_host=ic-monitor01 ssh_port=8888
```

因为组定义支持嵌套，所以我们可以预先定义一些组嵌套，并以 append 的方式追加到最终的 host.list 文件中。

```
[root@ic-admin2 batchRun]# cat config/host.list.append

[IC]
ETX_IC
CMP_IC

[EMU]
ETX_EMU
CMP_EMU

[root@ic-admin2 batchRun]# tools/switch_etc_hosts --input_file
config/hosts --output_file config/host.list --rewrite --tool
batchRun --append config/host.list.append

Output File : config/host.list
[root@ic-admin2 batchRun]# tail -n 10 config/host.list
[VM]
10.232.158.39 ssh_host=ic-monitor01 ssh_port=8888

[IC]
ETX_IC
CMP_IC

[EMU]
ETX_EMU
CMP_EMU
```

## 5.2 tools/save\_password

工具 save\_password 用于将当前用户密码加密保存。

```
[root@ic-admin2 batchRun]# tools/save_password -h
usage: save_password.py [-h] -p PASSWORD [-H HOST] [-o
OUTPUT_FILE]
```

```
optional arguments:
  -h, --help            show this help message and exit
  -p PASSWORD, --password PASSWORD
                        Specify user password.
  -H HOST, --host HOST  Specify the host which user password works
on, default is "default".
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Specify the output file, default is
"<db_path>/password/<user>".
```

**--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。

**--host**: 指定账号密码生效的机器，默认为所有机器。

**--password**: 指定用户明文密码。

下面是一个例子。

```
[root@ic-admin2 batchRun]# tools/save_password --password ***
[root@ic-admin2 batchRun]# cat db/password/root
root  default  1724049499  ed5fa5a1d09c8f4bf3e99c55708be23a
00b2a6273de7bc43a83f9bc352822827
```

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。

## 5.3 tools/sample\_host\_info

工具 sample\_host\_info 用于批量采集多服务器的 OS 和 hardware 信息，并保存为 json 文件备用。

其底层是借助了 batch\_run 的--output\_file 功能，所以会接受一些 batch\_run 的基本参数并传递给 batch\_run 执行，然后将 batch\_run 的输出信息整理后重新保存。

```
[root@ic-admin2 batchRun]# tools/sample_host_info -h
usage: sample_host_info.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-u USER] [-p PASSWORD] [-o OUTPUT_DIR]
```

```
optional arguments:
  -h, --help            show this help message and exit
  -H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
                        Specify the host(s) for batch_run.
  -G GROUPS [GROUPS ...], --groups GROUPS [GROUPS ...]
                        Specify host group(s) for batch_run.
  -u USER, --user USER  Specify the user name when connectting
host as.
  -p PASSWORD, --password PASSWORD
                        Specify the user password when connectting
host with.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Where to save temporary file and
host_info.json, default is current directory.
```

**--host**: 指定 host, “ALL”意味着 host.list 中所有的服务器。

**--groups**: 指定 group, “ALL”意味着 host.list 中所有的服务器, 于--host 参数二选一设置即可。

**--user**: 指定 user, 非必须。

**--password**: 指定 password, 非必须。

**--output\_dir**: 指定输出路径, 默认为当前路径。

下面是一个例子。

```
[root@ic-admin2 batchRun]# tools/sample_host_info --hosts ALL --
parallel 20 --output_dir db/host_info
>>> Move /ic/software/tools/batchRun/db/host_info/host_info.json
to
/ic/software/tools/batchRun/db/host_info/host_info.json.20241026
_132002 ...
mv /ic/software/tools/batchRun/db/host_info/host_info.json
/ic/software/tools/batchRun/db/host_info/host_info.json.20241026
_132002
>>> Clean up /ic/software/tools/batchRun/db/host_info ...
rm -rf *.info host_list.json host_info.json
>>> Sampling host list information ...
/ic/software/tools/batchRun/bin/batch_run --hosts ALL --list
--output_file
/ic/software/tools/batchRun/db/host_info/host_list.json
```

```
* Host(s) info has been saved into
"/ic/software/tools/batchRun/db/host_info/host_list.json".
>>> Sampling host os/cpu/mem information ...
    /ic/software/tools/batchRun/bin/batch_run --hosts ALL --
command "lsb_release -a; hostnamectl; lscpu; free -g" --parallel
20 --output_message_level 1 --output_file
/ic/software/tools/batchRun/db/host_info/HOST.info

Total 10 hosts.
>>> Collecting host information ...
    Host info has been saved to file
"/ic/software/tools/batchRun/db/host_info/host_info.json".
```

生成的 host\_info.json 文件内容如下。

```
{
    "10.232.136.12": {
        "server_type": "server",
        "os": "CentOS Linux release 7.9.2009 (Core)",
        "cpu_architecture": "x86_64",
        "cpu_thread": 48,
        "thread_per_core": 1,
        "cpu_model": "Intel(R) Xeon(R) Platinum 8260 CPU @
2.40GHz",
        "cpu_frequency": 3.1,
        "cpu_frequency_unit": "GHz",
        "mem_size": 1007,
        "mem_size_unit": "GB",
        "swap_size": 127,
        "swap_size_unit": "GB"
    },
    ...
}
```

## 5.4 tools/patch

工具 patch 是帮助 batchRun 打补丁的工具，其帮助信息如下。

```
[root@ic-admin2 batchRun]# tools/patch -h
usage: patch.py [-h] [-p PATCH_PATH]
```

```
optional arguments:  
  -h, --help            show this help message and exit  
  -p PATCH_PATH, --patch_path PATCH_PATH  
                        Specify patch path (new install package  
path).
```

**--patch\_path**: 指定补丁包（也就是新的安装包）路径。

一般而言，batchRun 的版本变更，主要是新增 python 脚本，或者是现有 python 脚本内容变更，针对这种变更的安装包，patch 可以自动将新的安装包变更更新到当前安装包。

下面是一个示例。

```
[root@ic-admin2 batchRun]# tools/patch -p ~/batchRun-master  
Install Path : /ic/data/usr/liyanqing.1987/tools/batchRun  
Patch path : /home/liyanqing.1987/batchRun-master  
  
*Warning*: current install path name is "batchRun", but patch path  
name is "batchRun-master".  
Do you want to continue? (y|n) y  
  
> Copying python file "/home/liyanqing.1987/batchRun-  
master/bin/batch_run.py" into  
"/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run.py".  
> Copying python file "/home/liyanqing.1987/batchRun-  
master/common/common.py" into  
"/ic/data/usr/liyanqing.1987/tools/batchRun/common/common.py".  
> Copying python file "/home/liyanqing.1987/batchRun-  
master/tools/encrypt_python.py" into  
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/encrypt_python.p  
y"
```

## 六、技术支持

本工具为开源工具，由开源社区维护，可以提供如下类型的技术支持：

- 部署和使用技术指导。
- 接收 bug 反馈并修复。
- 接收功能修改建议。(需审核和排期)

获取技术支持的方式包括：

- 通过 Contact 邮箱联系开发者。
- 添加作者微信 “liyanqing\_1987”，注明“真实姓名/公司/licenseMonitor”，由作者拉入技术支持群。



## 附录

### 附 1. 变更历史

日期	版本	变更描述
2022.12	1.0	发布第一个正式 release 版本
2023.7	1.1	增加对 host_ip 和 host_name 的多对多映射关系的支持。 去除对 LSF queue 机器获取的支持。
2024.8	1.2	增加服务器信息采样功能。
2024.10	2.0	增加 GUI 界面。