

日志收集功能—Linux

1. 修订

版本	内容	负责人	日期
0.1	方案设计说明，开发与测试报告	严汝宁	2022/11/03

2. 功能概述

本文主要介绍在Linux下获取相关日志文件的功能，其中通过想要获取文件的路径，在按一定条件筛选出来并解析输出文件后，一并打包然后通过调用redfish上传日志的相关接口上传到BMC内存中(存放路径:/var/tmp/hi_log)。

3. 目的

功能特性需求，收集系统日志。

4. 功能实现

4.1 开发环境

Ubuntu18.04、VSCODE、Python 3.6.9、pip 9.0.1等

4.2 pip模块

```
1 pip3 install filesplit
2 pip3 install requests
3 pip3 install pyinstaller
```

4.3 设计与实现

大致工作步骤如下：

- 1 sudo/root权限请求
- 2 参数选项的设计，包含IP、取日志内容的大小范围内等
- 3 获取日志文件
- 4 打包与压缩文件

- 5 redfish接口设计与实现
- 6 文件上传与结果解析
- 7 使用pyinstaller打包成可执行程序

4.4 代码（略）

...

4.5 redfish接口设计与实现

- 1 代码(略)

...

- 2 API接口

请求url: https://{ip}/redfish/v1/Oem/Powerleader/LogData

请求头:

参数	说明
Content-Type	multipart/form-data
X-Server-Addr	服务器usb0的IP。默认为169.254.0.17:。冒号不可缺省

请求体:

参数	说明
log	上传的文件 (类型:file)

响应体:

参数	说明
log_len	文件的长度, 单位bytes

响应示例:

```
1 | {"log_len":xxx}
```

4.6 参数选项的设计

本程序设计包含三个参数选项说明如下: (注意大小写与参数的有效性)

参数选项	说明
-ip xxx.xxx.xxx.xxx	USB链路IP地址。注：若此参数选项不指定ip,则默认为168.254.0.17
-MB xxx	取单个文件数据大小，单位MB。注：若此参数选项若不指定MB,则默认为1MB
-debug	调试输出使用。注：默认不打开，参数选上并执行之后会在本路径下生成debug.log

注：三者可以顺序可调，亦可只指定其中某一项。

注意: 168.254.0.17为BMC的usb0的默认ip,因此OS下要确保ping 168.254.0.17能通畅。

5. 测试与运行

5.1 开发中调试

在默认情况下，即没有参数选项，例如：

```
1 | sudo python3 ./logCollection.py
```

或者在root下

```
1 | python3 ./logCollection.py
```

加上三个参数选项之后，例如：

```
1 | sudo python3 ./logCollection.py -ip 192.168.43.234 -day 1 -debug
```

指定某个参数选项之后，例如：

```
1 | sudo python3 ./logCollection.py -ip 192.168.43.234
```

5.2 使用pyinstaller打包可执行程序

```
1 | pyinstaller -F ./logCollection.py
2 | # -i ICONNAME: 设置图标
3 | # -F: 表示将所有的程序全部打包在一起，生成的只有一个可执行文件(存放于dist)
```

命令执行之后会生成两个文件夹：一个叫build，另一个叫dist。其中生成之后build文件可不要。

最终在dist文件夹下生成logCollection可执行二进制文件。

5.3 运行可执行程序

注意：执行之前，请先赋予其可执行权限。

因获取日志有权限问题，sudo或者root权限运行。比如logCollection可执行文件在路径/home/yimning/dist/下。

terminal 下输入命令和相应参数并运行即可，如：

```
1 yimning@Ubuntu:~/dist$ sudo ./logCollection xxx xxx
2 # 其中xxx xxx 为参数选项
```

6. Linux下开启定时任务（重点）

6.1 Linux下查看定时任务

```
1 cat /etc/crontab
```

```
1 sudo cat /etc/crontab
```

内容如下：

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

对crontab内容说明如下：

```
1 # Example of job definition:
2 # .----- minute (0 - 59)
3 # | .----- hour (0 - 23)
4 # | | .----- day of month (1 - 31)
5 # | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
6 # | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
   sun,mon,tue,wed,thu,fri,sat
7 # | | | | |
8 # * * * * * user-name  command to be executed
9 # m h dom mon dow
10 # 分 时 天 月 周
```

其中

minute：表示分钟，可以是0到59之间的任何整数。

hour：表示小时，可以是0到23之间的任何整数。

day：表示日期，可以是1到31之间的任何整数。

month：表示月份，可以是1到12之间的任何整数。

week：表示星期几，可以是0到7之间的任何整数，这里的0或7代表星期日。

command：要执行的命令，可以是系统命令，也可以是自己编写的脚本文件。

在以上各个字段中，还可以使用以下特殊字符

星号 (*)：代表所有可能的值，例如month字段如果是星号，则表示在满足其它字段的制约条件后每月都执行该命令操作。

逗号 (,)：可以用逗号隔开的值指定一个列表范围，例如，“1,2,5,7,8,9”

中杠 (-)：可以用整数之间的中杠表示一个整数范围，例如“2-6”表示“2,3,4,5,6”

正斜线 (/)：可以用正斜线指定时间的间隔频率，例如“0-23/2”表示每两小时执行一次。同时正斜线可以和星号一起使用，例如*/10，如果用在minute字段，表示每十分钟执行一次。

注意:定时任务的触发时间由用户定义。

例如现在有定时任务如下：

```
1 */1 * * * * root sudo /home/yimning/dist/logCollection
2
```

表示每隔1分钟去执行这个 `sudo /home/yimning/dist/logCollection` 命令。

```
1 1 */1 * * * * root sudo /home/yimning/dist/logCollection -ip 192.168.43.234
  -debug
```

表示每隔1小时1分钟去执行这个 `sudo sudo /home/yimning/dist/logCollection -ip 192.168.43.234 -debug` 命令。

6.2 Linux开启定时任务

1 收集日志功能几个文件说明如下：(不可删)

```
1 project
2 |— disableCrontab.sh      # 取消定时化任务脚本
3 |— enableCrontab.sh      # 开启定时化任务脚本
4 |— logCollection          # 收集日志文件打包与上传
5 |— logFileDir             # 要收集日志文件的路径
6 |— README.md             # 文档说明
7
```

2 对enableCrontab.sh脚本部分内容的说明

```
1 # 参数选项说明：
2 # -ip xxx.xxx.xxx.xxx   IP地址      注：参数选项若不指定ip,则默认为168.254.0.17
3 # -MB xxx               数据大小 单位默认为MB      注：参数选项若不指定MB,则默认为1MB
4 # -debug                调试输出用      注：默认不打开，若此参数选上并执行之后会在本
  路径下生成debug.log
5 # 启动任务内容填写，例如：
6 # CONTENT="*/1 * * * * root sudo $CURRENT_PATH/$SHELL_TASK -ip
  192.168.43.234 -MB 1 -debug"
7 #（解析：每一分钟执行一次任务；ip为192.168.43.234；-MB:取每个日志文件的大小不超过1MB；-
  debug,启用调试输出）
8 #
9 # 默认值为：CONTENT="*/1 * * * * root sudo $CURRENT_PATH/$SHELL_TASK "
```

注意:定时任务的触发时间由用户定义，各需求不一样。

可在enableCrontab.sh脚本里面修改 `CONTENT="*/1 * * * * root`

`sudo $CURRENT_PATH/$SHELL_TASK"`

例如:让它每隔5分钟执行一次，修改如下:

```
1 | CONTENT="*/5 * * * * root    sudo $CURRENT_PATH/$SHELL_TASK"
```

例如:让它每隔12小时执行一次，修改如下:

```
1 | CONTENT="* */12 * * * root    sudo $CURRENT_PATH/$SHELL_TASK"
```

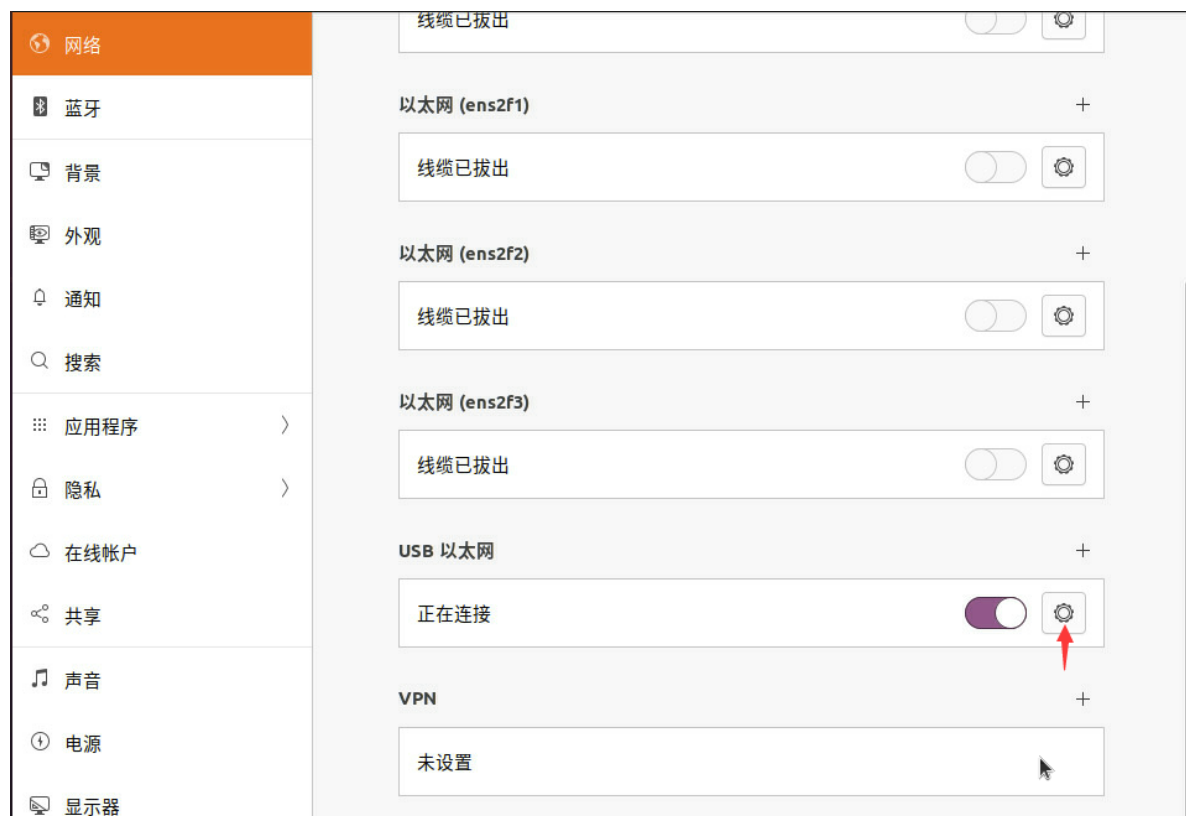
以此类推，不再详细介绍。

3 确保OS系统下的USB链路ip能与BMC的usb0 ip网络畅通可用。在OS下使用命令 `ping 168.254.0.17` 去检查，如网络ping不通，需要设置OS下的USB链路ip。

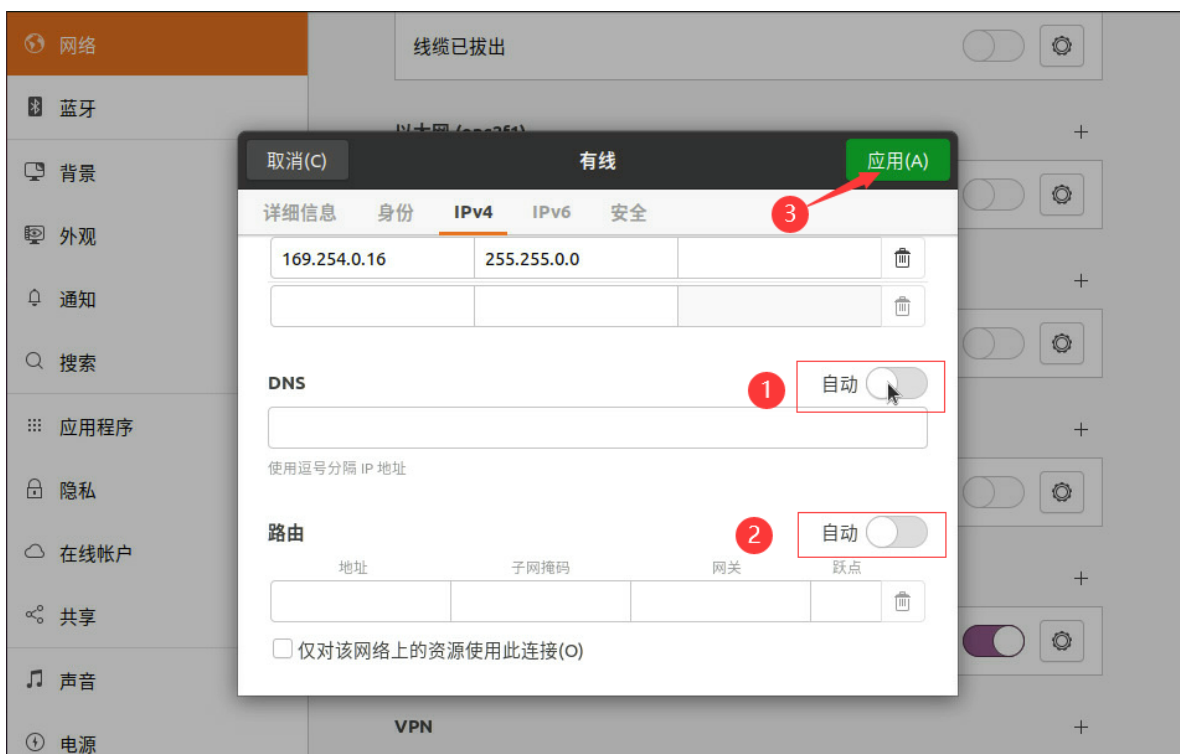
linux(以ubuntu 20.04为例)下配置USB以太网，如下:

找到设置---网络---USB以太网

带点击齿轮进行配置，如下:



配置IPV4，ip地址可配成169.254.0.xx,子网掩码：255.255.0.0,网关可暂不配置。详细配置如下:



点击应用之后，使用命令 `ping 169.254.0.17`，若能ping通即可。

```
jerry@jerry-ArcherCityM:~$ ping 169.254.0.17
PING 169.254.0.17 (169.254.0.17) 56(84) bytes of data.
64 比特, 来自 169.254.0.17: icmp_seq=1 ttl=64 时间=0.629 毫秒
64 比特, 来自 169.254.0.17: icmp_seq=2 ttl=64 时间=0.409 毫秒
64 比特, 来自 169.254.0.17: icmp_seq=3 ttl=64 时间=0.494 毫秒
64 比特, 来自 169.254.0.17: icmp_seq=4 ttl=64 时间=0.442 毫秒
64 比特, 来自 169.254.0.17: icmp_seq=5 ttl=64 时间=0.416 毫秒
64 比特, 来自 169.254.0.17: icmp_seq=6 ttl=64 时间=0.293 毫秒
```

4 执行开启定时化任务脚本如下：

注意：执行之前，请先赋予其可执行权限，如 `chmod +x xxx.sh`。因获取日志有权限问题，`sudo`或者`root`权限运行

一般在默认情况下，不带参数选项。除非有特殊需求才会指定参数。

```
1 | sudo ./enableCrontab.sh
```

加上三个参数选项之后，例如：

```
1 | sudo ./enableCrontab.sh -ip 192.168.43.234 -MB 1 -debug
```

指定参数选项-ip xxx，例如：

```
1 | sudo ./enableCrontab.sh -ip 192.168.43.234
```

指定参数选项-MB xxx，例如：

```
1 | sudo ./enableCrontab.sh -MB 1
```

指定参数选项-debug，例如：

```
1 | sudo ./enableCrontab.sh -debug
```

执行之后可用 `sudo cat /etc/crontab` 查看是否存在。

5 执行取消定时任务脚本如下：

```
1 | sudo ./disableCrontab.sh
```

6.3 测试结果

1 测试日志是否获取成功

在到该任务运行时间到了之后，在可执行程序的路径下(如：/home/yimning/dist/)见到多出os文件夹或者os.tar.gz压缩包(其中之一)即可。

```
yimning@Ubuntu:~/dist$ ls
disableCrontab.sh  enableCrontab.sh  logCollection  logFileDir  os  os.tar.gz
yimning@Ubuntu:~/dist$
```

2 测试文件包是否上传成功

在执行开启定时任务脚本即(enableCrontab.sh)的时候，添加参数，如下：

```
1 | sudo ./enableCrontab.sh -debug
```

或者

```
1 | sudo ./enableCrontab.sh -ip 192.168.43.234 -debug
```

添加参数选项-debug后并执行(注意有空格)。执行之后可用 `sudo cat /etc/crontab` 查看内容是否写入成功。

加上-debug和定时任务自动执行之后，在当前路径下生成debug.log文件。然后查看它输出的最新内容。如果见有

输出类似如下内容即表示上传成功,可看status_code是否为200，是否存在{"log_len":xxx}"等内容。


```
1 Tue Nov 15 16:31:02 2022 : ping 192.168.43.234 is ok, network is available.
2
3 Tue Nov 15 16:31:02 2022 : status_code: 200,upload_result: Upload
succeeded,response_result: {"log_len":xxx}
```

注意:最后如果不调试了，把参数选项-debug去掉。

上传失败的情况如下图示：

```
Tue Nov 15 16:15:01 2022 : status_code: 503,upload_result: Upload failed,response_result: <html><body>redfish out of service</body></html>

Tue Nov 15 16:15:01 2022 : -----This debug output end-----

Tue Nov 15 16:16:02 2022 : target_ip: 192.168.43.234, data_size_MBytes: 1, is_debug_mode: 1

Tue Nov 15 16:16:02 2022 : ping 192.168.43.234 is ok, network is available.

Tue Nov 15 16:16:02 2022 : status_code: 503,upload_result: Upload failed,response_result: <html><body>redfish out of service</body></html>

Tue Nov 15 16:16:02 2022 : -----This debug output end-----

Tue Nov 15 16:17:02 2022 : target_ip: 192.168.43.234, data_size_MBytes: 1, is_debug_mode: 1

Tue Nov 15 16:17:03 2022 : ping 192.168.43.234 is fail, please try again after network repair.

Tue Nov 15 16:17:03 2022 : The network is abnormal or the filePath of /home/yinning/yinning/app/python/dist/os.tar.gz does not exist. The program
exit(0).

Tue Nov 15 16:17:03 2022 : -----This debug output end-----
```

上传成功的情况如下图示：

```
Tue Nov 15 16:23:02 2022 : target_ip: 192.168.43.234, data_size_MBytes: 1, is_debug_mode: 1

Tue Nov 15 16:23:02 2022 : ping 192.168.43.234 is ok, network is available.

Tue Nov 15 16:23:03 2022 : status_code: 403,upload_result: Upload failed,response_result:

Tue Nov 15 16:23:03 2022 : -----This debug output end-----

Tue Nov 15 16:24:02 2022 : target_ip: 192.168.43.234, data_size_MBytes: 1, is_debug_mode: 1

Tue Nov 15 16:24:02 2022 : ping 192.168.43.234 is ok, network is available.

Tue Nov 15 16:24:02 2022 : status_code: 200,upload_result: Upload succeeded,response_result: {"log_len":52478}

Tue Nov 15 16:24:02 2022 : -----This debug output end-----

Tue Nov 15 16:25:01 2022 : target_ip: 192.168.43.234, data_size_MBytes: 1, is_debug_mode: 1

Tue Nov 15 16:25:01 2022 : ping 192.168.43.234 is ok, network is available.

Tue Nov 15 16:25:02 2022 : status_code: 200,upload_result: Upload succeeded,response_result: {"log_len":52493}

Tue Nov 15 16:25:02 2022 : -----This debug output end-----
```

上传失败原因分析：

- 没有使用管理员权限。
- 网络不通，ip设置有误。（常见）

3 WEBUI下载OS日志

“登录WEBUI——PL诊断——OS信息收集日志——点击一键下载OS相关信息”，点击下载即可。见下图操作所示：

