Python服务部署指南

Yonghong Z-Suite — V8.5.1

YonghongTech
Talk with Data

目录

1 Yonghong-DM-Engine-V1.0.0 简介

1.1 功能特点

Yonghong-DM-Engine 是永洪科技自研数据挖掘引擎，目前为第一个版本（V1.0.0），暂时仅用于为永洪产品 Yonghong Z-Suite 的深度分析功能提供相关支持。其本身通过远程过程调用（RPC）为永洪产品提供服务，可独立部署于本地或者远程服务器。Yonghong-DM-Engine 采用 Protobuffer 与 Yonghong Z-Suite 交换数据，数据传输效率比 JSON、XML 更高，尤其适用于大批量数据的交换。

1.2 系统需求

DM-Engine 在常见的 64 位服务器操作系统上均可部署，如 Window Server、Centos7.x、Redhat、Ubuntu 等。

注意：Yonghong-DM-Engine-V1.0.0 不支持在 32 位操作系统上安装。

对应的 Yonghong Z-Suite 版本：V8.5.1，不支持老版本。

Python 版本：3.6.x

1.3 文档内容说明

本文档不涉及 Yonghong Z-Suite 产品的安装及配置过程，这部分的资料请参阅永洪科技官网支持中心的相关教程（https://www.yonghongtech.com/zc/）。

由于用户的服务器系统环境多种多样，比如 Windows、Linux、Docker 的，服务器还有联网的、网络隔离的。本文档从第 2 章至第 5 章各对应了一种安装需求，各章内容完全独立，请根据自己的安装需求查阅对应的章节即可。

如果部署过程中遇到问题，请先查阅第 6 章的常见问题及解决方法。如果仍不能解决，请联系永洪技术支持。

2 联网 Linux 服务器上的部署过程

2.1 摘要：

为简化操作，Yonghong-DM-Engine-v1.0.0 的启动脚本 Yonghong-DM-Engine-v1.0.0/bin/run.sh 已经内置了 Python3.6.1 的安装过程，如果您的服务器系统为 CentOS7 版本且没有安装 python3，那么可以跳过以下第 2.2、2.3 节，按照第 2.4 节和第 2.5 节的操作配置并启动软件。

如果不满足以上系统环境，那么部署过程主要分为 4 个步骤：

Python3 安装--->所需工具包的安装--->DM-Engine 的配置--->启动 DM-Engine

以下各小节为每个步骤的操作细节。

2.2 Python 安装

必备步骤：

(1) 用 yum install xxx 命令下载 python3 的依赖

```
yum install zlib-devel bzip2-devel openssl-devel ncurses-devel epel-release gcc gcc-c++ xz-devel readline-devel gdbm-devel sqlite-devel tk-devel db4-devel libpcap-devel libffi-devel
```

(2) 下载 python3.6.1 安装包

在 python 官网下载（https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz）所需的 python3.6.1，或者使用 wget 命令下载

```
wget --no-check-certificate https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz
```

(3) 安装 python3.6.1

```
tar -zxvf Python-3.6.1.tgz # 解压 python3 安装包
cd Python-3.6.1 # 进入 python3 安装包目录
./configure --prefix=/usr/local/bin/python3.6.1 # 将 python3 安装在这个目录
su root # 编译安装前必须是 root 权限，或者使用 sudo 提权安装
```

```
make && make install # 编译和安装
```

(4) 创建软连接

```
ln -s /usr/local/bin/python3.6.1/bin/python3 /usr/bin/python3 # 创建 python3 软连接
ln -s /usr/local/bin/python3.6.1/bin/pip3 /usr/bin/pip3 # 创建 pip3 的软连接
exit # 程序安装完毕，如果此时为 root 角色，在启动程序前先退出 root 角色
```

至此 python3 和 pip3 的安装已经完成，接下来就是利用 pip3 来安装项目开发中所需的 python3 模块了。

2.3 工具包安装与更新

工具包的安装与更新推荐使用 pip3 命令进行安装，该工具可在线或者离线安装指定的工具包。

(1) 工具包批量安装

DM-Engine 依赖于一批指定的工具包，所需工具包及其版本已经在 requirements.txt 文件中列出，requirements.txt 位于 DM-Engine 软件包的根目录中。

首先执行命令：

```
mkdir ./ Yonghong-DM-Engine-v1.0.0 # 在 Yonghong-DM-Engine-v1.0.0.tar.gz 所在目录创建子目录
tar -zxvf Yonghong-DM-Engine-v1.0.0.tar.gz -C ./ Yonghong-DM-Engine-v1.0.0 # 解压到 Yonghong-DM-Engine-v1.0.0 目录
cd ./Yonghong-DM-Engine-v1.0.0 # 进入到 Yonghong-DM-Engine-v1.0.0 目录
```

再执行：

```
python3 -m pip3 install -r ./requirements.txt
```

批量安装各种工具包就行了。

如果安装过程中报错，往往是 linux 系统缺少模块所需的某些依赖工具，只要用 yum install 下载相应的依赖就行。

(2) 工具包更新

当需要安装或更新某工具包时，直接使用如下命令即可：

```
python3 -m pip3 install -U xxxx    # xxxx 是你要安装的工具包名
```

2.4 软件配置

在运行 DM-Engine 之前，必须先进行必要的配置。配置文件路径为 Yonghong-DM-Engine-v1.0.0/conf/Sysconfig.ini，其中可配置项如下图所示：

```
[installation]
library=%(app_path)s/bin/DeployEnv/Lib
include=%(app_path)s/bin/DeployEnv/include
bin=%(app_path)s/bin
app_path=../
# Setting related to debug configuration.
[debug]
log_errors=true
show_warnings=false
show_runtime_info=false
[server]
host=0.0.0.0
port=8099
# authorized options:true,false.
authorized=false
username=yonghongtech
password=123456
# nworkers had better to be set the value equals or smaller than the quantity of CPU cores.
nworkers=auto
# Setting the max cache of a RPC message, and the value is between 1MB and 2048MB.
max_cache=2000MB
PMML-copyright= Copyright (c) 2012-2019 YonghongTech

root=../
signature=YonghongTech---Talk with Data.
```

用户可配置项，
通常保持默认值即可

配置项说明：

host：主机名，本机 ip 地址，通常保存默认设置 0.0.0.0 即可，这样可以接受所有的内网和外网请求；

port：服务端口号,范围[0, 65535]，如果启动时提示 port 被占用，请更换一个端口号，重启服务即可；

authorized：指明客户端与服务建立连接时是否需要认证，true 表示客户端必须先使用用户名密码认证，false 表示不需要认证；

username：用户名，服务端需要认证时使用，可自行更改；

password：密码，服务端需要认证时使用，可自行更改；

nworkers：指定服务允许的最大并发数，一般并发数不宜超过 CPU 逻辑核数，当值为 auto 时最大并发数等于 CPU 核数，也可以根据需要设置为其它大于 0 的整数；

max_cache：消息请求的最大数据量，单位是 MB，可设置为 1MB 至 2048MB 之间的值，该项与数据表的行数密切相关，值越大单个消息可传输的数据行数越多，例如设置为 400MB 对应最大数据行数约为 100 万行（具体行数与每个数据的字节数有关）；

PMML-copyright：声明 PMML 文件的版权，会内置到 PMML 文件中，用户可以将版权改为指定的公司；

root：指定工作目录，取默认值即可，建议不要修改；

signature：签名，用于加密，建议不要修改。

2.5 DM-Engine 启动步骤

切换到 Yonghong-DM-Engine-v1.0.0/bin 目录，执行如下命令即可启动：

```
sh run.sh    #执行此命令的当前工作目录必须为 bin，否则软件无法运行
```

3 离线 Linux 服务器上的部署过程

3.1 摘要

很多用户的服务器处于内网中工作，从公网无法访问，而且服务器本身也不能访问外网，要在这种服务器上部署 Yonghong-DM-Engine-V1.0.0 工具，需要用户先在同版本操作系统的联网服务器中将对应的工具下载下来，然后拷贝到离线服务器中进行本地安装配置。

主要步骤：

（1）联网环境下载编译工具；

（2）联网环境下载 python 及其依赖包；

（3）将联网环境下载的所有工具包拷贝至离线服务器中；

（4）在离线服务中编译安装所有工具包；

（5）配置 DM-Engine；

（6）启动 DM-Engine 服务；

详细的步骤请参考下面两个小节。

3.2 在联网 linux 环境下载所需资源

详细步骤：

（1） 通过修改联网服务器上的 yum 配置，在使用 yum install xxx 的时候直接将这些依赖保存在指定目录中，而不是直接删除，便于后面将这些依赖包拷贝到离线服务器中。具体命令行操作如下：

```
vim /etc/yum.conf        # 打开 yum 配置文件
```

做如下修改：

```
cachedir=/var/cache/yum      # yum install xxx 中 xxx 依赖的存储位置
keepcache=1      # 改为 keepcache=1 这样 yum install xxx 下载的 rpm 包就不会删除，如果为 0 则安装完后不保留缓存
```

（2） yum install xxx 下载 python3 的依赖，具体命令行操作如下：

```
yum install zlib-devel bzip2-devel openssl-devel ncurses-devel epel-release gcc gcc-c++ xz-devel readline-devel gdbm-devel sqlite-devel tk-devel db4-devel libpcap-devel libffi-devel
```

（3） 下载 python3.6.1 安装包

在 python 官网下载（https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz）所需的 python3.6.1，或者使用 wget 命令下载：

```
wget  --no-check-certificate  https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz
```

（4） 安装 python3.6.1

```
tar -zxvf Python-3.6.1.tgz -C Python-3.6.1 # 解压 python-3.6.1 安装包
```

```
cd Python-3.6.1 # 进入 python3 安装包目录

./configure --prefix=/usr/local/bin/python3.6.1 # 将 python3 安装在这个目录

su root    # 编译安装前必须是 root 权限

make && make install # 编译和安装
```

（5） 创建软连接

```
ln -s /usr/local/bin/python3.6.1/bin/python3 /usr/bin/python3 # 创建 python3 软连接

ln -s /usr/local/bin/python3.6.1/bin/pip3 /usr/bin/pip3# 创建 pip3 的软连接

exit    # 程序安装完毕，如果此时为 root 角色，在启动程序前需要先退出 root 角色
```

至此 python3 和 pip3 的安装已经完成，接下来就是利用 pip3 来安装项目开发中所需的 python3 模块了。

（6） 批量下载所需的第三方包

```
pip3 download -r requirements.txt -d /tmp/packages/
```

然后将 requirement.txt 文件和/tmp/packages/下的这些模块都拷贝到离线服务器。

（7） 将下载好的 Python-3.6.1.tgz 和/var/cache/yum 目录下的所有 rpm 包都拷贝至离线服务器。

接下来是对离线服务器的操作。

3.3 在离线 linux 服务器中的运行环境配置

详细步骤：

（1） 安装 python3 的编译环境

在拷贝来的 rpm 包目录执行如下命令：

```
rpm -Uvh ./*.rpm --nodeps --force
```

（2） 安装 python3

```
tar -zxvf Python-3.6.1.tgz -C Python-3.6.1    # 解压安装包

cd Python-3.6.1    # 进入 python3.6.1 目录

./configure --prefix=/usr/local/bin/python3.6.1    # 将 python3 安装在这个目录
```

```
su root    # 编译安装前必须是 root 权限，如果已经是 root 权限，此步骤可跳过
make && make install    # 编译和安装
```

（3） 创建软连接

```
ln -s /usr/local/bin/python3.6.1/bin/python3 /usr/bin/python3    # 创建 python3 软连接
ln -s /usr/local/bin/python3.6.1/bin/pip3 /usr/bin/pip3    # 创建 pip3 的软连接
exit    # 程序安装完毕，如果此时为 root 角色，在启动程序前需要先退出 root 角色
```

至此 python3 和 pip3 的安装已经完成，接下来就是利用 pip3 来安装项目开发中所需的 python3 模块了。

（4） 批量安装所需模块（第三方包）

假设./tmp/packages 是在离线服务器中这些模块的存储位置，根据实际情况修改。

```
python3 -m pip3 install --no-index --find-links=./tmp/packages -r ./requirements.txt    # 拷贝过来的文件
```

3.4 DM-Engine 软件配置

在运行 DM-Engine 之前，必须先进行必要的配置。将 Yonghong-DM-Engine-V1.0.0.tar.gz 拷贝到离线服务器中，执行以下命令：

```
mkdir ./ Yonghong-DM-Engine-v1.0.0 # 在 Yonghong-DM-Engine-v1.0.0.tar.gz 所在目录创建子目录
tar -zxvf Yonghong-DM-Engine-v1.0.0.tar.gz -C ./ Yonghong-DM-Engine-v1.0.0 # 解压 Yonghong-DM-Engine-v1.0.0.tar.gz 到 Yonghong-DM-Engine-v1.0.0 目录
cd ./Yonghong-DM-Engine-v1.0.0 # 进入到 Yonghong-DM-Engine-v1.0.0 目录
```

配置文件路径为 Yonghong-DM-Engine-v1.0.0/conf/Sysconfig.ini，其中可配置项如下图所示：

```
[installation]
library=%(app_path)s/bin/DeployEnv/Lib
include=%(app_path)s/bin/DeployEnv/include
bin=%(app_path)s/bin
app_path=../
# Setting related to debug configuration.
[debug]
log_errors=true
show_warnings=false
show_runtime_info=false
[server]
host=0.0.0.0
port=8099
# authorized options:true,false.
authorized=false
username=yonghongtech
password=123456
# nworkers had better to be set the value equals or smaller than the quantity of CPU cores.
nworkers=auto
# Setting the max cache of a RPC message, and the value is between 1MB and 2048MB.
max_cache=2000MB
PMML-copyright= Copyright (c) 2012-2019 YonghongTech

root=../
signature=YonghongTech---Talk with Data.
```

用户可配置项，
通常保持默认值即可

配置项说明：

host：主机名，本机 ip 地址，通常保存默认设置 0.0.0.0 即可，这样可以接受所有的内网和外网请求；

port：服务端口号,范围[0, 65535]，如果启动时提示 port 被占用，请更换一个端口号，重启服务即可；

authorized：指明客户端与服务建立连接时是否需要认证，true 表示客户端必须先使用用户名密码认证，false 表示不需要认证；

username：用户名，服务端需要认证时使用，可自行更改；

password：密码，服务端需要认证时使用，可自行更改；

nworkers：指定服务允许的最大并发数，一般并发数不宜超过 CPU 逻辑核数，当值为 auto 时最大并发数等于 CPU 核数，也可以根据需要设置为其它大于 0 的整数；

max_cache：消息请求的最大数据量，单位是 MB，可设置为 1MB 至 2048MB 之间的值，该项与数据表的行数密切相关，值越大单个消息可传输的数据行数越多，例如设置为 400MB 对应最大数据行数约为 100 万行

（具体行数与每个数据的字节数有关）；

PMML-copyright：声明 PMML 文件的版权，会内置到 PMML 文件中，用户可以将版权改为指定的公司；

root：指定工作目录，取默认值即可，建议不要修改；

signature：签名，用于加密，建议不要修改。

3.5 启动 DM-Engine 服务

命令行窗口切换到 Yonghong-DM-Engine-v1.0.0/bin 目录，执行如下命令即可启动：

```
sh run.sh    #执行此命令的当前工作目录必须为 bin，否则软件无法运行
```

4 使用 Docker 镜像部署过程

4.1 摘要

为方便已经安装了 Docker 的用户部署 DM-Engine，永洪科技为用户提供了 DM-Engine 的安装镜像文件，您导入镜像（使用命令：`docker load < Yonghong-DM-Engine-V1.0.0.tar`）后只需要按照下面第 4.4 节的命令启动即可。

Docker 镜像版本如下：

Docker Version: 18.09.6
Docker API version: 1.39 (minimum version 1.12)
OS: centos7(kernal minimum version 3.10)

如果镜像文件不适合您的系统或者您想自行创建镜像，我们还提供了 dockerfile 文件供您参考使用。从 dockerfile 文件部署的主要步骤：

（1）准备工作，主要是安装 Docker；

（2）根据 dockerfile 文件构建 DM-Engine 镜像；

（3）创建并启动 DM-Engine 容器；

（4）配置 DM-Enginge 应用。

以下各小节为每个步骤的操作细节。

4.2 准备工作

假设你的系统是 centos7，且已配置了 docker，此步跳过。如果还没有安装 docker 请依次使用以下命令安装配置（docker 官方文档：https://docs.docker.com/install/linux/docker-ce/centos/）：

```
yum install -y docker    #安装 docker
systemctl stop firewalld.service    #关闭防火墙
systemctl start docker.service    #启动 docker
systemctl enable docker.service    #设置开机自启动
```

4.3 创建 DM-Engine 镜像

详细步骤：

（1）在 CentOS 系统中解压 DM-Engine 软件，在解压文件目录中会看到有 dockerfile 文件。

```
tar -zxvf Yonghong-DM-Engine-V1.0.0.tar.gz ./dm-engine-v1.0.0
```

（2）进入到 dockerfile 文件所在目录 dm-engine-v1.0.0。

```
cd ./ dm-engine-v1.0.0
```

（3）执行命令：

```
docker build --tag= dm-engine-v1.0.0 .
```

该命令将创建名为 dm-engine-v1.0.0 的镜像。

4.4 创建并启动 DM-Engine 容器

使用命令：

```
docker run -p 8099:8099 dm-engine-v1.0.0
```

即可启动应用程序，dm-engine-v1.0.0 容器和内部应用的端口号均为 8099。

4.5 配置 DM-Engine

默认应用的默认配置一般不需要修改即可运行，如果要修改某些配置，请参考以下详细步骤：

（1）使用命令

```
docker cp ID:/yh_pyserver/conf/SysConfig.ini .   #将 ID 改成你的容器 ID
```

（注意上面命令尾部有个点）把 SysConfig.ini 文件复制到当前目录下；

（2）根据需求修改 SysConfig.ini 的配置内容，配置项说明如下：

```
[installation]
library=%(app_path)s/bin/DeployEnv/Lib
include=%(app_path)s/bin/DeployEnv/include
bin=%(app_path)s/bin
app_path=../
# Setting related to debug configuration.
[debug]
log_errors=true
show_warnings=false
show_runtime_info=false
[server]
host=0.0.0.0
port=8099
# authorized options: true, false.
authorized=false
username=yonghongtech
password=123456
# nworkers had better to be set the value equals or smaller than the quantity of CPU cores.
nworkers=auto
# Setting the max cache of a RPC message, and the value is between 1MB and 2048MB.
max_cache=2000MB
PMML-copyright= Copyright (c) 2012-2019 YonghongTech

root=../
signature=YonghongTech---Talk with Data.
```

用户可配置项，
通常保持默认值即可

配置项说明：

host：主机名，本机 ip 地址，通常保存默认设置 0.0.0.0 即可，这样可以接受所有的内网和外网请求；

port：服务端口号,范围[0, 65535]，如果启动时提示 port 被占用，请更换一个端口号，重启服务即可；

authorized：指明客户端与服务建立连接时是否需要认证，true 表示客户端必须先使用用户名密码认证，false 表示不需要认证；

username：用户名，服务端需要认证时使用，可自行更改；

password：密码，服务端需要认证时使用，可自行更改；

nworkers：指定服务允许的最大并发数，一般并发数不宜超过 CPU 逻辑核数，当值为 auto 时最大并发数等于 CPU 核数，也可以根据需要设置为其它大于 0 的整数；

max_cache：消息请求的最大数据量，单位是 MB，可设置为 1MB 至 2048MB 之间的值，该项与数据表的行数密切相关，值越大单个消息可传输的数据行数越多，例如设置为 400MB 对应最大数据行数约为 100 万行（具体行数与每个数据的字节数有关）；

PMML-copyright：声明 PMML 文件的版权，会内置到 PMML 文件中，用户可以将版权改为指定的公司；

root：指定工作目录，取默认值即可，建议不要修改；

signature：签名，用于加密，建议不要修改。

（3）使用命令

```
docker cp ./SysConfig.ini ID: /yh_pyserver/conf/SysConfig.ini     #将 ID 改成你的容器 ID
```

把修改后的配置文件传回容器中，替换旧的配置文件。

（4）修改容器内应用程序的配置文件后必须重启容器，使用命令：

```
docker restart ID     #将 ID 改成你的容器 ID
```

（5）如果你想查看容器内 Yonghon-DM-Engine 的标准输出，使用命令：

```
docker logs -f ID     #将 ID 改成你的容器 ID
```

5 Windows 系统上的部署过程

5.1 摘要

DM-Engine 在 windows 系统上的部署比较简单，python3 解释器已经内置到软件包中并预先安装了常用的第三方包，直接将软件包解压到指定目录按照第 5.4 节的步骤启动即可。剩下的工作就是第三方工具包的安装与更新、启动前的软件配置。

5.2 工具包安装与更新

工具包的安装与更新推荐使用 pip3 命令进行安装，该工具可在线或者离线安装指定的工具包。

当需要安装或更新某工具包时，使用如下命令即可：

```
python3 -m pip3 install -U xxxx     # xxxx 为你的包名
```

特别注意，不能直接使用 `pip3 install xxxx` 命令安装，该命令会报错或者将工具包安装到操作系统环境变量 path 指定的目录下。

5.3 软件配置

在运行 DM-Engine 之前，根据需要可进行必要的配置。

配置文件路径为 Yonghong-DM-Engine-v1.0.0/conf/Sysconfig.ini，其中可配置项如下图所示：

```
[installation]
library=%(app_path)s/bin/DeployEnv/Lib
include=%(app_path)s/bin/DeployEnv/include
bin=%(app_path)s/bin
app_path=../
# Setting related to debug configuration.
[debug]
log_errors=true
show_warnings=false
show_runtime_info=false
[server]
host=0.0.0.0
port=8099
# authorized options:true,false.
authorized=false
username=yonghongtech
password=123456
# nworkers had better to be set the value equals or smaller than the quantity of CPU cores.
nworkers=auto
# Setting the max cache of a RPC message, and the value is between 1MB and 2048MB.
max_cache=2000MB
PMML-copyright= Copyright (c) 2012-2019 YonghongTech

root=../
signature=YonghongTech---Talk with Data.
```

用户可配置项，
通常保持默认值即可

配置项说明：

host：主机名，本机 ip 地址，通常保存默认设置 0.0.0.0 即可，这样可以接受所有的内网和外网请求；

port：服务端口号,范围[0, 65535]，如果启动时提示 port 被占用，请更换一个端口号，重启服务即可；

authorized：指明客户端与服务建立连接时是否需要认证，true 表示客户端必须先使用用户名密码认证，false 表示不需要认证；

username：用户名，服务端需要认证时使用，可自行更改；

password：密码，服务端需要认证时使用，可自行更改；

nworkers：指定服务允许的最大并发数，一般并发数不宜超过 CPU 逻辑核数，当值为 auto 时最大并发数等于 CPU 核数，也可以根据需要设置为其它大于 0 的整数；

max_cache：消息请求的最大数据量，单位是 MB，可设置为 1MB 至 2048MB 之间的值，该项与数据表的行数密切相关，值越大单个消息可传输的数据行数越多，例如设置为 400MB 对应最大数据行数约为 100 万行（具体行数与每个数据的字节数有关）；

PMML-copyright：声明 PMML 文件的版权，会内置到 PMML 文件中，用户可以将版权改为指定的公司；

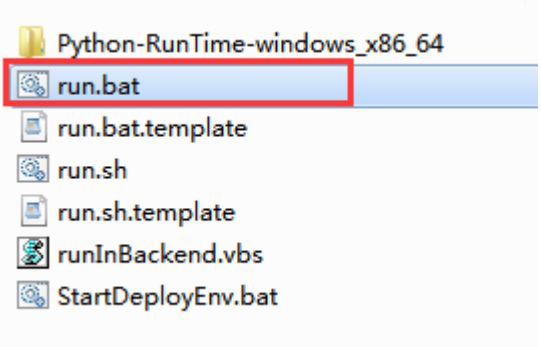
root：指定工作目录，取默认值即可，建议不要修改；

signature：签名，用于加密，建议不要修改。

5.4 DM-Engine 启动步骤

DM-Engine 在 windows 系统服务器上的启动过程非常简单：

在 Yonghong-DM-Engine-V1.0.0 的 bin 文件夹双击如下图的 run.bat 文件运行即可，或者在 run.bat 文件上鼠标右键选“打开”即可。



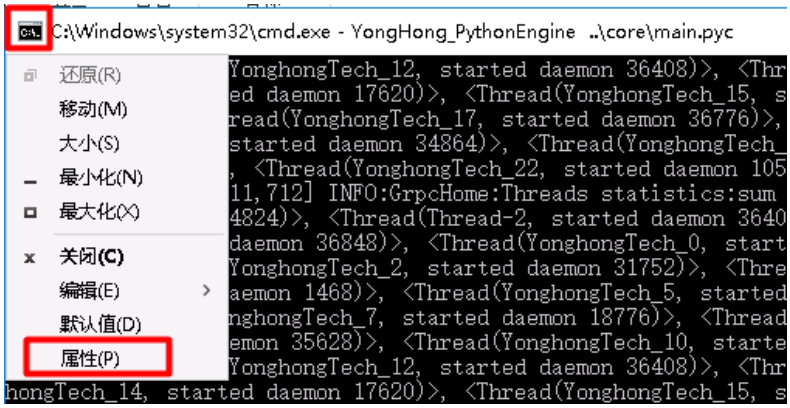
6 常见问题及解决方法

6.1 问题 1：在 Windows 系统中运行时，控制台窗口停止打印日志，程序卡住。

原因：windows 操作系统的 dos 界面默认设置为“快速编辑模式”导致鼠标点击控制台窗口的日志内容时进入“快速编辑模式”，对应的进程暂停（并未终止）以接受鼠标操作。

解决方法：此时用鼠标点击窗口内任意内容并鼠标右键单击，即可使程序继续运行。或者，按键盘“Enter”键也可以恢复正常。最佳方式是取消控制台的“快速编辑模式”，如下如所示：

第一步，选择属性；



第二步，取消勾选“快速编辑模式”，点击“确定”即可；

6.2 问题 2：在 linux 系统上执行命令 python3 -m pip3 install -r ./requirements.txt 批量安装库时报错。

这种情况往往是 linux 系统缺少模块所需的某些依赖工具或者编译工具。根据错误提示，用 yum install 下载安装缺失的工具，再重新执行 `python3 -m pip3 install -r ./requirements.txt` 即可。

6.3 问题 3：在 linux 系统上运行启动脚本 run.sh 时报错。

出错提示：

原因：

运行 run.sh 脚本时当前工作目录未在 Yonghong-DM-Engine-V1.0.0/bin，而程序入口文件以此目录作为相对目录。

解决方法：

一定要进入目录 Yonghong-DM-Engine-V1.0.0/bin 再启动 run.sh 脚本。