

# 程序使用说明

孙中阳 [szy@sunzhongyang.com](mailto:szy@sunzhongyang.com)

## 文档目标

---

帮助使用者在符合条件且配置好的硬件环境中运行程序，执行程序的各项功能

## 前期准备

---

### 配置环境

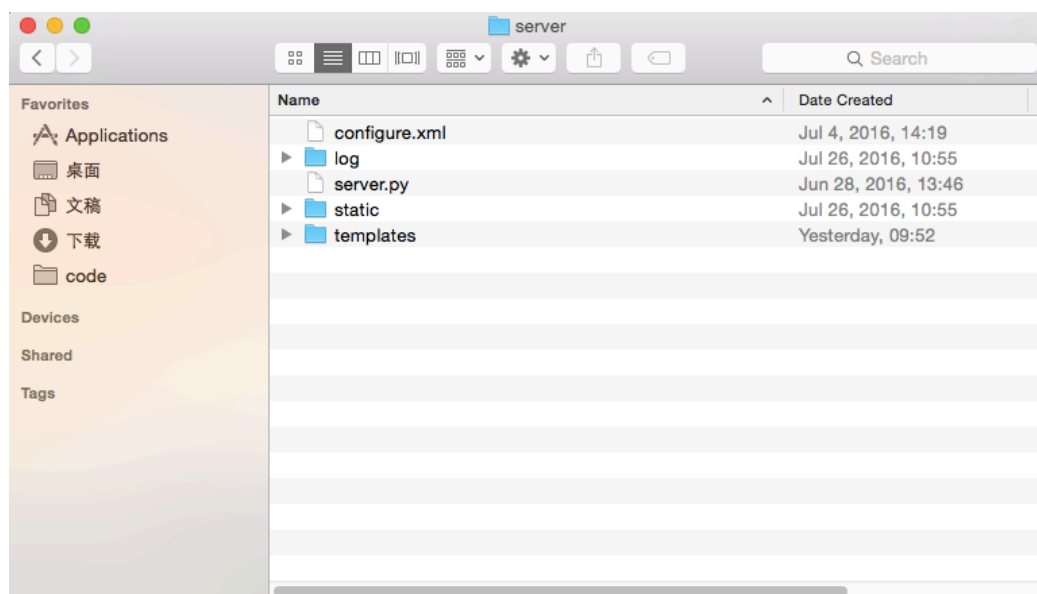
按照配置说明中有关内容配置合适的运行环境

## 安装程序

---

解压缩文件 `Log-Collect-Analysis.zip`

产生两个文件夹 `client`和`server`，`client`文件夹中的`client.py`是客户端程序，不多赘述。`server`文件夹中除服务端程序`server.py`之外还有如下文件



其作用分别如下

### **configure.xml**

服务端的配置文件，保存服务端的各项设置

### **log**

保存全部客户端发来的所有日志文件原件

### **static**

保存服务端程序生成的静态文件，如在网页中用于显示日志统计结果的图片文件，保存需要下载日志的压缩包等

### **templates**

保存Tornado用来渲染显示统计结果的网页的模板

## 安装客户端

将得到的client文件夹拷贝至客户端机器相关目录（任意合适目录均可）

## 安装服务器

将得到的server文件夹拷贝至服务器机器相关目录（任意合适目录均可）

## 配置程序

---

按以下步骤执行

### 配置服务端

打开configure.xml，更改默认配置，默认配置在default标签中

```
<configure self="2016.1.2">
<default>
  <nickname>default_name</nickname>
  <mail>szy@sunzhongyang.com</mail>
  <logpath>/home/szy/</logpath>
  <logname>[a-z]+[_][0-9]+(.log.)[0-9]{4}[-][0-9]{2}[-][0-9]{2}</logname>
</default>
</configure>
```

每当有一个新终端连接服务端后，若其之前没有连接过，则服务端会按default

中有关配置为其生成一个自动配置，生成后的文件如下所示

```
<configure self="2016.1.2">
<default>
  <nickname>default_name</nickname>
  <mail>szy@sunzhongyang.com</mail>
  <logpath>/home/szy/</logpath>
  <logname>[a-z]+[_][0-9]+(.log.){0-9}{4}[-][0-9]{2}[-][0-9]{2}</logname>
</default>
<client id="iz28tdw0507Z">
  <nickname>default_name</nickname>
  <mail>szy@sunzhongyang.com</mail>
  <logpath>/home/szy/</logpath>
  <logname>[a-z]+[_][0-9]+(.log.){0-9}{4}[-][0-9]{2}[-][0-9]{2}</logname>
</client>
</configure>
```

其中“id=”后为客户端机器的网络名称，这里为iz28tdw0507Z，其他各项配置分别为

### nickname

每台客户端的标识，更改后可标记不同客户端。同时报警邮件包含的客户端名即为此名称

### mail

mail后可以“:”分隔接多个邮件地址。当某客户端日志检测到FATAL错误时会为邮件地址列表中的所有地址发送邮件

### logpath

标记客户端机器上的日志地址，要求必须为绝对地址

### logname

正则表达式形式的日志文件命名规则。本例中为

`[a-z]+[_][0-9]+(.log.){0-9}{4}[-][0-9]{2}[-][0-9]{2}`，可检测如 `taskmgr_14412.log.2016-06-27` 的日志名

另外，如果启动服务端没有检测到congifure.xml，则其会自动生成一个和本例中初始configure.xml相同的configure.xml

## 配置客户端

找到client.py如下部分

```
#服务端hostname或者IP地址
hostname = 'sunzhongyang.com'
#服务端端口号
port = 12333
```

更改服务端地址为有关地址

## 配置支持服务

### MongoDB

找到server.py中如下部分

```
#建立和本地MongoDB数据库的连接，默认端口号为27017
connection = pymongo.MongoClient('localhost', 27017)
```

可通过更改此部分连接远程或者需要密码的MongoDB等

### Mail

找到server.py中如下部分,并修改

邮件服务器设置

```
mail_host="smtp.***.***" #设置服务器
mail_user="***" #用户名
mail_pass="***" #口令
```

发件人名称

```
sender = 'szy@sunzhongyang.com'
```

邮件报警功能需要连接邮件服务器才能使用。根据测试163会拦截邮件内容中形似代码的部分，所以最好使用自己搭建的邮件服务器，

## 启动程序

---

### 启动服务端

切换shell到服务端所在目录，输入命令

```
python server.py --port=12333
```

--port=后接服务端程序监听的端口号，端口号在server.py中按如下方式指定

```
#设置服务端端口号 (default=端口号)
from tornado.options import define, options
define("port", default=12333, help="run on the given port",
    type=int)
```

本例中，将端口号指定为12333

## 启动客户端

切换shell到客户端所在目录，输入命令

```
python client.py
```

如果需要定时启动客户端，则可使用crontab命令，具体使用方法如下  
首先切换到拥有足够权限的用户输入

```
crontab -e
```

进入crontab的配置界面,加入如下配置

```
$HOME=client.py所在目录
1 * * * * python client.py
```

即可使client.py在每小时的01分自动启动并运行。  
\$HOME=client.py所在目录 是必需的，否则client.py在执行时会找不到有关文件的位置

## 查看统计结果

---

程序成功启动后可登陆网页查看统计结果

首先打开<http://localhost:端口号/check>，出现如下页面

Alpha

Week

## Search

From date

To date

---

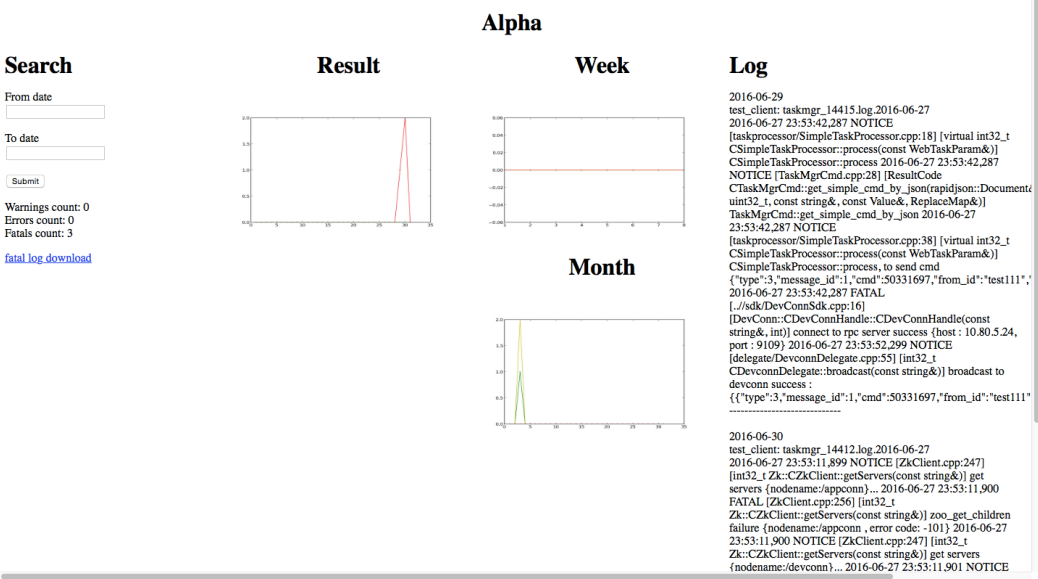
然后可在Serach中输入日志生成时间的范围，搜索该范围内的日志

## Search

From date

To date

点击Submit，跳转至详情页面



其中Result图表保存查询结果视图，Week和Month则是周视图和月视图。详情网页左侧支持继续搜索，且可下载全部范围内出现FATAL错误的日志的全文。右侧Log显示出现FATAL错误的日志的有关详细信息，包括具体时间等。另外也可通过curl获取静态网页文件，不多赘述。

## 自动报警邮件

服务端会检查收到的日志文件，如果发现日志中有FATAL错误，则向必要的地址发送邮件

**log\_analysis\_server@iZ28tdw0507Z**   
To: server\_administrator@iZ28tdw0507Z  
A FATAL error occurred

July 5, 2016 at 17:09



a FATAL error occurred in test\_client



2016-07-05 17:09 taskmgr\_14412.log.2016-06-27

**log\_analysis\_server@iZ28tdw0507Z**   
To: server\_administrator@iZ28tdw0507Z  
A FATAL error occurred


July 5, 2016 at 17:09



a FATAL error occurred in test\_client



2016-07-05 17:09 taskmgr\_14415.log.2016-06-27

**log\_analysis\_server@iZ28tdw0507Z**   
To: server\_administrator@iZ28tdw0507Z  
A FATAL error occurred

July 5, 2016 at 16:54



邮件内容包括出现FATAL错误的客户端的nickname，发件时间和错误日志的  
文件，可下载查看