南華大學
UNIVERSITY OF SOUTH CHINA

密　级：普通

文档编号：20170630-SRS　　版　次：2

分册名称：　　第 1 册/共 1 册

分布式爬虫

（20170630）

安装部署说明书

编　　写　：　杨奎 王浩茂

校　　对　：　杨奎

审　　核　：　刘立

批　　准　：　刘立

生效日期　：　2017 年 6 月 30 日

南华大学计算机学院制

2017 年 6 月

文件修改控制

修改记录编号	修改状态	修改页码及条款	修改人	审核人	批准人	修改日期

目录

1. 引言

1.1目的

本文档描述了运行“分布式爬虫”程序的系统运行环境的构建，其中包括依赖软件的安装过程以及配置过程，和系统本身的部署过程。

1.2背景

爬虫系统，是对海量的分散的互联网数据进行采集的系统，是搜索引擎系统的基础。大数据近年来快速发展，炙手可热，不仅是数据的容量大，更是强调对全样本的数据的分析。互联网数据中包含了大量有价值信息，是大数据的重要数据来源。

而互联网上的数据内容丰富，组织形式也灵活多样。传统的爬虫系统，对所有的网页采用同样的办法处理，利用深度优先或广度优先的办法获取网页链接，下载网页，对网页中的所有的文本数据建立倒排索引。这种方式没有对网页数据的信息进行组织、归类。

应大数据的需求，分布式爬虫系统是解决这一问题的方案。分布式爬虫，对同一个网站的同类数据，进行结构化。同时，能利用分布式的软件设计方法，实现爬虫的高效采集。

2. 特别说明

项目已部署到腾讯云服务器，如果本地不需要部署，则可以直接访问这里进行使用。

3. 系统运行环境

- 软件环境： Ubuntu 14.04 及以上
- 数据库：MongoDB、Redis
- 应用服务器：Apache2
- 硬件环境：CPU 为酷睿双核、网络 100M、存储器应采用 SCSI 高速硬盘且容量应大于 30G，数据备份可采用磁盘阵列 Raid5，应用服务器部署到物理服务器 A。（非最低配置环境）

4. 系统运行环境的搭建

4.1服务器的安装配置

- 系统的服务器端运行在 Ubuntu 平台。
- Apache2 的安装过程参见 这里 。

4.2数据库的安装配置

由于本系统采用的是 MongoDB 数据库和 Redis 数据库。

- MongoDB Enterprise 的安装过程参见官方文档。
- Redis 的安装过程参见官方文档。

5. 部署系统

5.1依赖包安装

```
sudo apt-get install libpq-dev python-dev libxml2-dev
libxslt1-dev libldap2-dev libsasl2-dev libffi-dev
```

5.2Python 模块安装

```
Scrapy: pip install scrapy
Django: pip install django
scrapy-redis: pip install scrappy-redis
requests: pip install requests
beautifulsoup: pip install beautifulsoup4
pymongo: pip install pymongo
redis:pip install redis
psutil: pip install psutil
```

5.3Graphite 安装及部署

- 参考资料：

http://www.open-open.com/lib/view/open1419683400953.html
http://blog.csdn.net/hjhmpl123/article/details/53967823

http://tripleday.cn/2016/10/06/graphite/

- 步骤 1：安装 graphite

```
pip install whisper
pip install carbon
pip install graphite-web
```

依赖:

```
sudo apt-get install libpq-dev python-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-dev libffi-dev
```

安装 graphite-web 后缺少 manage.py

解决办法：

从 https://github.com/graphite-project/graphite-web.git 下载 graphite-web，将其中的 manage.py 拷贝到/opt/graphite/webapp 下

- 步骤 2：配置 graphite

1.graphite 会自动安装在/opt/graphite 目录下

将/opt/graphite/conf 下的*.example 去掉.example 后缀

```
sudo mv aggregation-rules.conf.example aggregation- rules. conf
sudo mv blacklist.conf.example blacklist.conf
sudo mv carbon.conf.example carbon.conf
sudo mv carbon.amqp.conf.example carbon.amqp.conf
sudo mv dashboard.conf.example dashboard.conf
sudo mv graphTemplates.conf.example graphTemplates.conf
sudo mv relay-rules.conf.example relay-rules.conf
sudo mv rewrite-rules.conf.example rewrite-rules.conf
sudo mv storage-aggregation.conf.example storage-aggregatio n .conf
sudo mv storage-schemas.conf.example storage-schemas.conf
sudo mv whitelist.conf.example whitelist.conf
suso mv graphite.wsgi.example graphite.wsgi
```

2.修改/opt/graphite/webapp/graphite/local_settings.py

将 local_settings.py.example 修改为 local_settings.py

sudo mv local_settings.py.example local_settings.py

SECRET_KEY = ''#使用一个长序列的字符串来代替默认的。可用 sha 等

TIME_ZONE = 'Asia/Shanghai'#这里需要修改成上海的时间，默认是美国芝加哥的时间

DEBUG = True#开启 dubug，这样就浏览器预览的时候，会查看到错误。

GRAPHITE_ROOT = '/opt/graphite'

3.修改/opt/graphite/conf 下 storage-aggregation.conf

```
[scrapy_min]
pattern = ^scrapy\..*_min$
xFilesFactor = 0.1
aggregationMethod = min
[scrapy_max]
pattern = ^scrapy\..*_max$
xFilesFactor = 0.1
aggregationMethod = max
[scrapy_sum]
pattern = ^scrapy\..*_count$
xFilesFactor = 0.1
aggregationMethod = sum
```

- 步骤 3：配置 scrapy 的 setting.py

```
STATS_CLASS = 'scrapygraphite.GraphiteStatsCollector'
GRAPHITE_HOST = 'ip'
GRAPHITE_PORT = 2003
```

- 步骤 4:启动 graphite

启动 carbon，carbon 会在默认的 2003 端口接收数据。

python /opt/graphite/bin/carbon-cache.py start

异常：

```
AttributeError: 'module' object has no attribute 'OP_NO_TLSv1_1'
```

解决：使用低版本 twisted

```
sudo pip uninstall twisted
sudo pip install twisted==13.1.0
```

启动 django，即整个 Graphite 的 web 应用。

```
python /opt/graphite/bin/carbon-cache.py start
```

可以指定端口启动：

```
python manage.py runserver 0.0.0.0:8085
```

异常：启动之后对数据库进行同步可能会出现报错，no such table:account_profile

解决：使用 python manage.py migrate --run-syncdb 命令进行同步

异常：

```
ImportError: No module named scandir
```

解决：

```
sudo pip install scandir
```

异常：

```
dlopen() failed to load a library: cairo / cairo-2
```

解决：

```
sudo apt-get install libpango1.0-0
sudo apt-get install libcairo2
 sudo apt-get install libpq-dev
```

之后用浏览器访问将看到一下界面：http://localhost:8000/

- 步骤 5：scrapy 启动前要运行以下代码：

```
sys.path.append('/opt/graphite/webapp/')
os.environ.setdefault("DJANGO_SETTINGS_MODULE",
graphite.settings")
```

➢ 步骤 6：在 apache2 上部署

安装 apache2：

```
sudo apt-get install apache2
```

拷贝文件：

```
sudo cp examples/example-graphite-vhost.conf /etc/apache2/sites-available/
example-graphite-vhost.conf
```

文件需要修改内容如下：

```
 Listen 8085
WSGISocketPrefix /var/run/apache2/wsgi
<VirtualHost *:8085>
   ServerName graphite
   DocumentRoot "/opt/graphite/webapp"
   ErrorLog /opt/graphite/storage/log/webapp/error.log
   CustomLog /opt/graphite/storage/log/webapp/access.log common
   WSGIScriptAlias / /opt/graphite/conf/graphite.wsgi
Alias /content/ /opt/graphite/webapp/content/
   <Location "/content/">
   SetHandler None
   </Location>
...
<VirtualHost>
```

到此，Graphite 部署成功，通过 http://ip:port 可以访问。

➢ 补充：

1.清理 graphite 中的 scrapy 数据

2.删除/opt/graphite/storage/whisper 下的 scrapy 文件夹即可

5.4爬虫管理系统部署

在 settings 文件中需要设置 LOCAL_HOST 为本机公网 ip 地址，若爬虫全部部署在一个局域网上，在设置为内网 ip 地址。

➢ 步骤 1：控制后台（网站的部署）

1. 环境与 web 服务器的安装

运行一下 shell 脚本:

```
#升级
sudo apt-get update
#安装 Apache2 服务器
sudo apt-get install Apache2
#安装 pip，一般 ubuntu14.04 以后的版本自带 python，不需要安装
sudo apt-get install python-pip
sudo pip install --upgrade pip #升级 pip
#安装 django
pip install Django
#如运行时错误，查看 Django 版本，若为版本问题，可更换 django 版
本
pip install Django==1.11
#建立 Python 与 Apache 的链接，安装 libapache2-mod-wsgisudo apt-get
install libapache2-mod-wsgi
#Python2 环境
```

2. 配置文件

```
#将 Django 工程放在/var/www/
sudo cp project /var/www/
#修改配置文件
sudo vim /etc/apache2/sites-available/geowind_crawler.conf
geowind_crawler.conf 的详细页面在下面给出
#配置文件生效
sudo a2ensite geowind_crawler.conf
#重启 Apache。
sudo service apache2 restart
```

3. geowind_crawler.conf 配置文件

```
<VirtualHost *:80>
ServerName 123.207.230.48
#ServerAlias otherdomain.com
#ServerAdmin youremail@gmail.com
# 存放用户上传图片等文件的位置
```

```
Alias /media /var/www/project/geowind_crawler/crawlermanage/media/
# 静态文件(js/css/images)的存放位置
#【注】: 切记写成 Alias /static/ /var/www/myweb/weibo/static/
Alias /static /var/www/project/geowind_crawler/crawlermanage/static/

# 允许通过网络获取 static 的内容
<Directory /var/www/project/geowind_crawler/crawlermanage/static>
        Require all granted
</Directory>
WSGIScriptAlias                                                    /
/var/www/project/geowind_crawler/geowind_crawler/wsgi.py
<Directory /var/www/project/geowind_crawler/geowind_crawler/>
<Files wsgi.py>
        Require all granted
</Files>
</Directory>
</VirtualHost>
```

4. 修改 wsgi.py

```
import os
import sys
from django.core.wsgi import get_wsgi_application
from os.path import join,dirname,abspath
PROJECT_DIR = dirname(dirname(abspath(__file__)))
sys.path.insert(0,PROJECT_DIR)
os.environ.setdefault("DJANGO_SETTINGS_MODULE",
"geowind_crawler.settings")
application = get_wsgi_application()
```

- 步骤 2：爬虫进程管理类的部署

进程管理类是一个 python 的 .py 文件，单独在后台运行。此处使用 screen 在回台运行。文件路径同步骤 1。

1. 安装 screen

```
$ sudo apt-get install screen
```

2. 后台运行

```
$ screen -S pymanage#这样新建一个名字为 pymanage 的窗口，并进入该窗口中
    $ python /var/www/project/bigcrawler/geospider/control/manage.py
    按 control + a +d 返回原界面
```

6. 下一步

至此，部署完成。你可以通过在浏览器键入

[IP：端口号]/crawlermanage/ 进行访问。