Gflux系统部署

北京网感至察信息科技有限公司

2015年03月09日

1. 运行环境安装

操作系统

此系统要求运行于Ubuntu 12.0.4 LTS操作系统之上，要求硬件服务器具有2GB以上内存和60GB以上的数据盘空间。

必备软件库与语言环境

在缺省安装上述操作系统之后，运行python命令检查语言版本，必须为2.7.3以上、3.0以下版本。

用如下命令安装必须软件库：

sudo apt-get install build-essential,gearmand,memcached,postgresql

然后用如下命令安装Python库：

sudo pip install django==1.6 gearman python-memcached django-localeurl sqlalchemy psycopg2 python-dateutil flup xlrd

sudo yum install gcc-gfortran libgfortran blas-devvel blas lapack lapack-devel gcc-c++ libuuid-devel gperf boost-devel memcached memcached-devel freetype-devel libpng-devel

sudo pip install numpy

sudo pip install scipy matplotlib scikit-learn

1. 服务器程序安装

服务器程序软件包为gflux.tar.gz，将其在Linux系统解压缩的命令如下：

tar zxvf gflux.tar.gz

假定解压缩之后程序包的路径为：/home/user/gflux/，里面应含manage.py文件和gflux子目录。称该路径为GFLUX\_DIR。

1. 服务器配置与启动

假定服务器所在的域名为XXX.COM, 服务器程序被部署之后的URL地址为[http://xxx.com/](http://abc.yunji.com.cn/yunji/)gflux/，称xxx.com为GFLUX\_DOMAIN。

配置nginx

在nginx的配置文件中加入：

upstream gflux\_fastcgi {

server 127.0.0.1:9091 weight=1 max\_fails=3 fail\_timeout=300s;

}

server {

………………………………..

location /yunji {

fastcgi\_pass gflux\_fastcgi;

fastcgi\_param PATH\_INFO $fastcgi\_script\_name;

fastcgi\_param REQUEST\_METHOD $request\_method;

fastcgi\_param QUERY\_STRING $query\_string;

fastcgi\_param CONTENT\_TYPE $content\_type;

fastcgi\_param CONTENT\_LENGTH $content\_length;

fastcgi\_param SERVER\_PROTOCOL $server\_protocol;

fastcgi\_param SERVER\_PORT $server\_port;

fastcgi\_param SERVER\_NAME $server\_name;

fastcgi\_param REMOTE\_ADDR $remote\_addr;

fastcgi\_param REMOTE\_PORT $remote\_port;

fastcgi\_pass\_header Authorization;

fastcgi\_intercept\_errors off;

}

………………………………..

}

此项配置将/gflux目录以fastcgi的模式导向本地的9091服务端口。

配置PostgreSQL数据库

sudo -u postgres psql

#create user tao with password 'pku123';

#alter role tao with superuser;

#create database gflux;

sudo service postgresql stop

sudo vi /etc/postgresql/9.\*/main/pg\_hba.conf

modify peer to md5 for local

modify add record host like below:

host all all 0.0.0.0/0 md5

sudo vi /etc/postgresql/9.\*/main/postgresql.conf

modify listen to \*

sudo service postgresql start

此项配置创建了一个数据库以及数据库用户，最后将数据库开放给所有ip段访问。

配置django

在local\_settings.py中配置PYTHON\_BIN，DATABASES和SQL\_BACKENDS，其中PYTHON\_BIN是python可执行文件的路径。

运行如下命令创建表格和初始化数据:

python manage.py syncdb

python manage.py init\_gflux\_dev\_env

启动服务器

启动django服务：

python collectstatic\_manage.py collectstatic --noinput

python manage.py runfcgi daemonize=True host=127.0.0.1 port=9091

启动nginx服务：

service nginx restart

使用supervise启动django服务器：

supervise是一个小巧的进程守护工具，它能在进程崩溃退出的时候自动重启进程，在启动时只需传入文件夹路径，supervise就能自动运行run文件，运行如下命令启动

nohup ./supervise run\_fcgi/ &

**添加计划任务**

运行命令 crontab crontab.conf

里面有如下的计划：

1. pre\_compute\_slow\_response 离线计算网感指数的脚本

数据初始化命令

采取如下方式往数据库中导入样本数据进行开发调试：

1. 启动django 服务器

2. 启动gearman\_worker\_import\_data.py

3. 打开上传网页，选择中油，使用data/all1000.txt 和 data/card1000.txt 上传

4. 点击开始导入

1. 使用与维护

**新增postgresql服务器**

1. 在新服务器上部署配置好postgresql服务器

2. 更改所有django的配置文件，在SQL\_BACKENDS添加新的服务器地址

3. 重启所有的django服务器

**新增django服务器**

1. 按照部署文档将django部署并配置好

2. 在新的服务器上启动django服务

3. 更改nginx的配置文件，在 gflux\_fastcgi 字段中添加新的服务器地址

往GFlux中导入新的中石油加油站数据

Windows:

1. 安装interbase\_xe3和marcketcloud
2. 将marcketcloud安装目录中的PositiveIBExternal.dll文件复制到interbase的安装目录中（Program Files\Embarcadero\InterBase\UDF）
3. 安装python（2.7）
4. 配置marcketcloud安装目录中的文件trans\_db\_settings.py，将各个文件或目录的路径改成自己真实的路径
5. 配置python环境变量
6. 将gdb文件放到配置过的data目录下，data目录中可以建多个文件夹来存放多个不同的gdb文件
7. 在命令行中进入marcketcloud安装目录，使用python运行trans\_db\_to\_all\_card.py文件。会将生成的文件在盘的根目录下打包。
8. 将生成的打包文件使用pscp上传到服务器

pscp H:\all\_card\_file.tar.gz [work@gflux1.tmlsystem.com](mailto:work@gflux1.tmlsystem.com):

Linux：

1. 登录服务器并解压缩文件：
2. tar xf all\_card\_file.tar.gz
3. 进入gflux目录 develop/gilbarco/trunk/gflux 运行命令导入数据：
4. python manage.py submit\_cnpc\_task --dir /home/work/data/ --user tao --location\_name=CN\_LN --location\_desc=辽宁

参数说明：

--dir 导入某个目录下的所有文件

--file 导入某个文件（需要写文件的完全地址）

--user 将数据导入某个用户中

--location\_name站点所在地的大写缩写，用下划线连接

--location\_desc 站点所在地的说明

--site\_name 导入站点的大写缩写，用下划线连接

--site\_desc 导入站点的说明

**往GFlux中导入新的中石化加油站数据**

1. 取得规定格式的txt数据文件（格式见附一）
2. 运行如下命令python manage.py submit\_senmei\_task --file='/home/work/develop/gilbarco/trunk/data/连潘.txt' --site='FZ\_LF' --site\_desc='FZ\_LF' --location\_name='BJ\_CN' --location\_desc='北京' –user='tao'

参数说明：

--file 给定文件的路径

--site 导入站点名

--site\_desc 导入站点的描述

--locatino\_name 导入站点的地点名

--location\_desc 导入站点的地点描述

**将导入的油站分配给指定用户使用**

1. 使用管理员登录，选择用户管理中的‘用户油站管理’
2. 在第一个下拉框中选择对方用户，点击查看可以看到对方用户现在能查看数据的油站。
3. 在第二个下拉框中选择油站，点击添加按钮可以将该油站的数据开放给对方用户。

附一：

单行范例涵义：

site+trans\_type+cardnum+payment\_type+timestamp+barcode+pay+quantity+desc+price+unitname+pump\_id+trans\_id

字段解释：

site 站点名

trans\_type 0为油品 1 为非油品

cardnum 卡号 默认为0

payment\_type 交易类型 1 银联卡 2 加油卡 3 信用卡 1000 现金

timestamp 交易时间 例如 2015-01-01 00：00：00：00

barcode 商品条形码

pay 消费金额

quantity 油品则为升数 非油品则为数量

desc 商品名

price 单价

unitname 商品单位 油品一般为公升

pump\_id 油枪号 非油品为0

trans\_id 交易流水编号

真实数据实例：

福州本部五一加油站,0,1000413500000035828,1000,2013-11-1 00:06:04,12827124,103.01,13.92,93号汽油（Ⅳ）,7.4,公升,007,11150001

福州本部五一加油站,0,1000413500000082222,1000,2013-11-1 00:07:54,9661572,150,18.96,97号汽油（Ⅳ）,7.91,公升,016,11150002