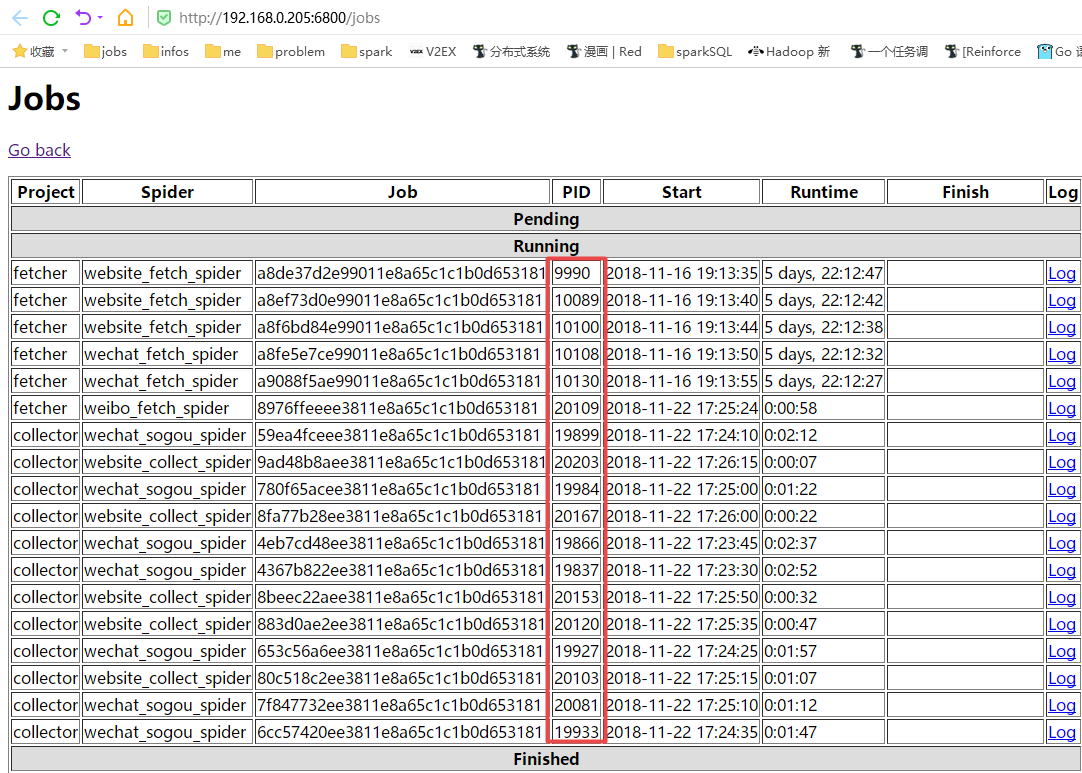
爬虫环境部署

## 总体运行情况

* 1. 目前爬虫系统部署在 **内网** 平台的 [192.168.0.205] [192.168.0.206] [192.168.0.207] 上
  2. 使用登陆账号为 ssc （Scrapy Spider Client），密码为 wislife\_205 、 wislife\_206 、 wislife\_207
  3. 所需代码均在 ssc 用户的 HOME 目录下
  4. 目前 Scrapy 分布式爬虫的发布机器为本机，可迁移
  5. 各虚拟环境 virtualenv 的作用：
     1. py2env —— Scrapy 分布式爬虫所需环境
     2. venv —— 计划任务脚本所需环境
     3. py3env —— （只存在 205 上）处理代理 IP 池脚本所需环境
     4. 以上环境在其环境目录下都包含了最新的依赖文件：requirements.txt，迁移环境的时候可以直接使用 **pip install -r requirements.txt** 重新构建各部署环境；
  6. 目前各主机均运行爬虫 collector 、 fetcher（py2env）以及 transform（venv），205 机器负责其余全部的定时脚本的运行
  7. 注意，每台机器的 transform 爬虫所负责的任务都不相同：
     1. 205 ： 负责处理微博数据进入 es
     2. 206 ： 负责处理网站数据进入 es
     3. 207 ： 负责处理微信数据进入 es
  8. 微信模拟器爬虫部署在 Windows 机器上
  9. 中移动招标网、顺企网肇庆黄页（定制爬虫）部署在 206 机器上，位于 huangjiahao 用户的 HOME 目录下

## 部署步骤

1. 如何重新发布 Scrapy 爬虫：
   1. 部署准备（迁移注意，一般情况下已准备）：运行的 worker 机器，其虚拟环境必须安装 scrapyd（**pip install scrapyd**）
   2. 先将 205 机器上的 ***dispatch\_fetcher\_spider.py*** 以及 ***dispatch\_spider.py*** 终止（ **ps -ef|grep python** 命令找到对应的 PID 然后 **kill -9**）
   3. 根据对应主机的 Scrapyd 部署状态 Web 页面先手动 **kill -9 PID** 全部对应的 pid 进程
   4. （切换到部署的 master 机器操作，本例子是本地机器）找到代码目录，首先确保部署 master 的机器的虚拟环境安装了 scrapy-deploy （**pip install scrapy-deploy**），同时，将虚拟环境的 bin（windows 下为 Scripts）目录下的 scrapyd-deploy 放到 ***代码目录/spider/collector*** 以及 ***代码目录/spider/fetcher*** 下
   5. 发布 collector —— 切换到虚拟目录，键入以下命令，利用 scrapyd 部署： **python scrapyd-deploy 205/206/207 -p collector** （Linux 机器上不需要指定用 Python 运行）
   6. 发布 fetcher —— 切换到虚拟目录，键入以下命令，利用 scrapyd 部署：**python scrapyd-deploy 205/206/207 -p fetcher** （Linux 机器上不需要指定用 Python 运行）
   7. 重启 205 机器上的脚本 ***dispatch\_fetcher\_spider.py*** 以及 ***dispatch\_spider.py*** ， **nohup python dispatch\_fetcher\_spider.py &** ， **nohup python dispatch\_spider.py &**
   8. \* 注意：目前设置是 8 点和 16 点会运行爬虫，所以其余时间段重新部署了暂时在 Web 页面上也看不到任何爬虫信息
   9. 启动 scrapy-splash 程序（206机器）： **docker run -d -p 8050:8050 scrapinghub/splash --max-timeout 300**
   10. 更新微博点赞数进程：（205 机器） **python <项目目录>/work/toolkit/weibo\_update/weibo\_update.py**
   11. 域名服务器部署（206机器）：**<项目目录>/ work/ws-domain/control.sh start**
2. 定制爬虫——中移动招标网
   1. 目前中移动招标网爬虫在每天的 8 点和 16 点都会生成报告发送到事业部同事的邮箱上，而爬虫则是每个小时都会执行一遍获取最新信息
   2. 爬虫的定时运行是通过 crontab 控制，目前使用的是 huangjiahao 用户的 crontab 计划，可以通过 **crontab -l** 查看
   3. 爬虫信息存到 206 数据库CMMC 上，对应的表为 bid
3. 定制爬虫——顺企网肇庆黄页（目前不再运行）
   1. 爬虫信息存放到 202 数据库 zhaoqing 上
   2. 需要运行爬虫，需先切换到对应虚拟环境 py27scrapy，然后切换到 shunqi/shunqi 目录下，然后键入以下命令 **scrapy crawl zhaoqing**
4. 微信模拟器爬虫部署

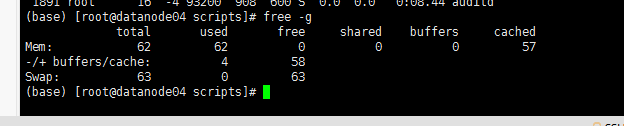
详见文档《微信模拟器爬虫部署》

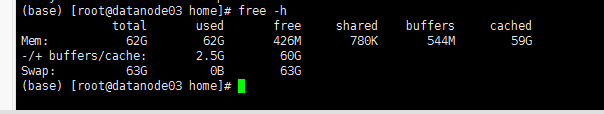
## 部署环境文件说明

1. Scrapy 爬虫（以 205 为例，位于 ssc 的 HOME 目录下）
   1. py2env 目录 —— Scrapyd 的爬虫虚拟目录
   2. py3env 目录 —— 代理IP池的虚拟目录
   3. venv 目录 —— 定时脚本的虚拟目录
   4. scrapyd —— 分发的爬虫目录
      1. dbs —— 不清楚
      2. eggs —— 编译为 eggs 文件的 Python 爬虫脚本
      3. logs —— 爬虫运行日志
      4. twisted.pid —— scrapyd 运行的 twisted 服务的 PID 文件
   5. software —— 所需额外的下载软件的目录（迁移环境需要）
   6. work —— 定时脚本所在目录
      1. scheduler —— 一般定时任务
         1. push\_queue.py —— （每小时执行）负责将 Web 端的配置读到 Redis 中，生成爬虫队列，运行： **nohup python push\_queue.py &**
         2. push\_setting.py —— （每半小时执行一次）将个人中心的配置数据（特指舆情动态）读到 Redis 中，运行： **nohup python push\_setting.py &**
         3. push\_num.py —— （长驻一直运行）更新微信和微博的点赞数到 es 中，运行： **nohup python push\_num.py &**
         4. daemon\_status.py —— （每5秒执行一次） 更新 scrapyd 客户端的状态信息到 redis 以便配置分发爬虫任务，运行： **nohup python daemon\_status.py &**
         5. push\_proxy\_ip.py —— （长住一直运行）更新代理 IP 到 redis ，运行： **nohup python push\_proxy\_ip.py &**
         6. dispatch\_fetcher\_spider.py —— （每分钟执行一次）监控分发 fetcher 爬虫，运行： **nohup python dispatch\_fetcher\_spider.py &**
         7. dispatch\_spider.py —— （长驻一直运行）派发爬虫任务服务，运行： **nohup python dispatch\_spider.py &**
      2. task —— 商情web 推送报告的定时任务
         1. push\_report\_task\_py —— 推送商情平台报告脚本 使用 **nohup python push\_report\_task\_py &** 运行
         2. update\_report\_task.py —— 更新商情平台报告推送配置脚本 使用 **nohup python update\_report\_task.py &** 运行
      3. transform —— 长驻的 Python 脚本任务
         1. application.py —— 运行的主文件，使用 **nohup python application.py &** 运行
         2. config.py 配置文件，可以修改数据库和 redis 链接
      4. toolkit —— 长驻进程，负责其他定时任务
         1. weibo\_update.py —— 负责将微博的转发数、点赞数和阅读数
2. 定制爬虫（206 机器，位于 huangjiahao 的 HOME 目录下）
   1. app 目录：代码目录
      1. cmmc —— 中移动招标网爬虫代码
      2. py27scrapy —— 虚拟环境目录
      3. report —— 中移动招标网的每日定时报告存储目录
      4. scripts —— 定时脚本目录
         1. build\_csv\_per\_day.\* —— （crontab 每天运行两次）生成报告并发送邮件
         2. run\_spider\_per\_hour.sh —— （crontab 每小时执行一次）定时运行中移动招标网爬虫
      5. shunqi —— 顺企网肇庆黄页爬虫代码
      6. weibo\_crawler —— 微博固定账号爬虫，负责爬取广州气象局相关微博的信息
   2. logs 目录：crontab 定时任务日志文件目录
3. 无

问题：爬虫会隔断时间不执行

原因：205/206服务器内存用完





方法是：

Sync（因为系统在操作的过程当中，会把你的操作到的文件资料先保存到buffer中去，因为怕你在操作的过程中因为断电等原因遗失数据，所以在你操作过程中会把文件资料先缓存。

所以我们在清理缓存先要先把buffe中的数据先写入到硬盘中，sync命令

）

echo 1 > /proc/sys/vm/drop\_caches

当然，这个文件可以设置的值分别为1、2、3。它们所表示的含义为：

echo 1 > /proc/sys/vm/drop\_caches:表示清除pagecache。

echo 2 > /proc/sys/vm/drop\_caches:表示清除回收slab分配器中的对象（包括目录项缓存和inode缓存）。slab分配器是内核中管理内存的一种机制，其中很多缓存数据实现都是用的pagecache。

echo 3 > /proc/sys/vm/drop\_caches:表示清除pagecache和slab分配器中的缓存对象。

Linux积极使用swap空间

echo 100 > /proc/sys/vm/swappiness

vim /etc/sysctl.conf

添加：

vm.swappiness=100