## 一、项目简介

       该项目是一套爬虫系统，目的是根据配置项爬取指定联盟的相关数据，其区别于 bdg\_outgoing/program/crawl 老系统，特点在于新系统为每次数据爬取分配了一个编号 crawl\_job\_id，并且记录每次数据爬取及处理（检测，同步）的状态于表 crawl\_job\_batch。不仅如此，新系统还将记录每个爬虫任务下的子爬虫的状态于表 batch 中，使得每次爬虫的每个数据爬虫版本的抓取状态可视化（详情见工具

二、新爬虫系统原理  
---------------  
#### 1. 项目背景  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
爬虫```bdg\_outgoing/program/crawl```老系统数据采取的过程中，默认每次爬取的数据都是正确数据，直接将爬下来的数据存入正式库，经常会有联盟返回数据的格式变化或因页面改版而导致爬取数据错误，   
因此常有脏数据进入正式系统，并且这样的情况发生我们甚至监控不到。其次是老爬虫系统对一些不敏感，而又多次重复爬取的大字段数据没有作防止重复爬取的操作，该部分数据占据了爬取数据的绝大部分，该部分数据消耗量大量的服务器资源，   
我们发现这些不便之后，迁移并改写了爬虫的服务至 ```br03``` 服务器上。  
#### 2. 爬虫原理  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;   
新爬虫的运行方式与老爬虫一致也是通过服务端配置定时脚本的方式启动总爬虫任务，定时启动脚本命令为：

php /home/bdg/crawl/crawl\_program\_jobs.php –method=getXXX crawl\_program\_jobs.php脚本会先从数据库中读取爬虫配置表aff\_crawl\_config表中的数据，得到需要爬取的联盟列表，然后遍历联盟列表依次并行执行具体联盟的爬虫程序。 受启于老爬虫系统的不足，新系统将需要爬取的联盟数据分为'实时更新数据'和'基础数据'，其中基础数据与账号无直接关联，同一联盟不同账号下该部分数据是相同的，并且这部分数据不经常改动。 因此，在爬取过程中新系统会给每次爬虫定义一个全量爬取的概念，当爬虫被设置为全量爬取时，系统会爬取'实时更新数据'和'基础数据'，并且'基础数据'会作防止重复爬取操作（在程序中控制是否缓存）；当爬虫被设置为非全量爬取时，系统仅会爬取'实时更新数据'。 其中'实时更新数据'存储于表batch\_program\_account\_site\_$networkId```中，'基础数据'存储于表```batch\_program\_$networkId中，其二者区分由这个两张表决定，因此，在新添爬虫时，请注意数据类型的区分，数据项应该与表中的数据项一一对应。 千万记住在写爬虫时，在做非全量爬取时，不要将'基础数据'做数据库更新，否则会被视为全量爬取，而向batch\_program\_$networkId```中插入部分数据，而导致爬虫检测为数据异常。爬取流程图如下图示： 数据爬取流程图

#### 3. 数据检测原理

       爬虫数据检测的流程与爬虫爬取类似，也是通过服务端配置定时脚本的方式启动总检测任务，系统每隔十分钟启动脚本：

php /home/bdg/crawl/crawl\_program\_jobs.php --method=checkXXX

其中单个联盟的检测流程如下图示：  
![数据检测流程图](http:api03.i.brandreward.com/crawl/check\_principle.png "数据检测流程图")  
#####   
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
评定数据准确性的原理为：找到最新一次爬虫任务，将这次任务获取的数据（存储于```batch\_program\_account\_site\_$networkId，batch\_program\_$networkId```中）与正式数据  
（存储于```program\_account\_site\_$networkId，program\_$networkId```中）做对比，若正式数据为空，默认新爬下的数据为正确数据，若是此次爬虫数据与正式数据差别较大（所有字段变化的总行数大于系统总行数的20%，或者正式数据中超过5%的数据在检测数据中未找到），则系统将会此版本数据设置为异常数据，  
并将异常数据的变化量生成log保存于```crawl\_job\_batch```表中的```CheckResult```字段中，并发送邮件通知技术人员，另外系统还会分析log内容将异常数据的字段名和主键保存于```CheckAnalyzeResult```字段中。  
系统选取检测数据时，会优先选取最新一次全量跑爬虫任务的数据为检测对象，如果检测通过系统会将该版本之前未检测的爬虫任务的状态设置为```Expired```。  
  
#### 4. 数据同步原理  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
爬虫数据同步的过程与检测的流程基本一样（见爬虫检测流程图），系统每隔十分钟启动脚本：

php /home/bdg/crawl/crawl\_program\_jobs.php –method=syncXXX crawl\_program\_jobs.php脚本会先从数据库中读取爬虫配置表aff\_crawl\_config表中的数据，得到需要爬取的联盟列表，然后遍历联盟列表依次并行执行具体联盟的同步程序。 同步的原理为：找到最近一次检测通过而又未同步的爬虫任务，在表batch\_program\_account\_site\_$networkId，batch\_program\_$networkId中，找到此次任务的数据，同步并覆盖至表program\_account\_site\_$networkId，program\_$networkId```中。

## 三、新爬虫系统正式数据访问

       新爬虫系统的数据都是通过API的方式向外输出，svn中代码的位置为：http://svn.meikaiinfotech.com/repo/bdg\_outgoing/crawl\_center/api； API的访问脚本位于br03的/app/site/api03.brandreward.com/web/crawl，各种类型的数据API入口都在该目录下。

#### \*API访问说明（以program数据为例）： #####请求链接地址：

http://api03.i.brandreward.com/crawl/program.php

#####参数说明：

params

required

default

description

key

YES

Not NULL

保存于department中的ApiKey,为部门使用该API的密钥

networkId

YES

Not NULL

表network主键,请求访问制定networkId联盟的数据

siteId

YES

Not NULL

表account\_site主键,请求访问对应联盟指定account\_site站点的数据

page

NO

1

分页页码

pagesize

NO

100

单次请求访问数据条数

## 四、[爬虫可视化工具](http://admin.brandreward.com/b_tools_crawl_center_status_display.php)

       爬虫可视化工具是为了监控和处理爬虫异常的后台管理工具，在该工具中我们可以预览最近七天的爬虫任务的异常 ([默认页](http://admin.brandreward.com/b_tools_crawl_center_status_display.php))， 也可以查看过去七天或者指定日期相关联盟所有爬虫任务的详情（[搜索页](http://admin.brandreward.com/b_tools_crawl_center_status_display.php)）， 也可以查看每次爬虫任务的具体情况和日志（[详情页](http://admin.brandreward.com/b_tools_crawl_center_detail.php?crawljobid=53749)）。另外，在爬虫任务的详情页，我们还可以查看检测异常任务所得数据的异常情况，若是数据变化量太多，工具能直观罗列出数据变化具体情况，这时需要人工判别是否此次检测异常情况，确实是爬虫源头大批量数据改变，亦或是爬虫异常。 当确定是爬虫源头大批量数据改变，若爬取格式并未发生变化，数据确实是我们想要的，我们可以在工具中手动设置爬取数据结果的状态为DONE，来将新改变数据导入正式数据系统；若是爬下来数据格式不对，并不是我们所需要的数据，这时基本可以判定爬取源头页面改版，届时需要修改对应爬虫。 另外，在详情页给出了爬虫日志文件保存的位置，以及日志文件预览功能，我们可以很方便点击查看爬虫日志内容。

## 五、过期数据删除

       由于新系统记录每次爬取的数据，海量的过期无用的重复数据就会占用大量存储空间，所以过期数据的删除是非常必要的。 过期数据删除的脚本(clearup.php)位于crawl\_center的根目录，数据清理设置为服务器时间每天的12点，清理详情参见br03上的日志文件/home/bdg/logs/clearup.log。 过期数据的清理规则是：只保留最近七天爬虫生成的缓存文件；爬下来的版本数据，保留最近90天的全部版本数据，90天以前的版本数据只保留在batch\_XXX\_changelog中留有记录的数据。

## 六、项目总结

       新爬虫系统很大程度上解决了脏数据直接入正式库的问题，并且使得每一次爬取的数据有迹可循，给技术人员运营数据带来了很大的便利。 由于笔者水平有限，系统存在很大的优化空间，例如：防重爬缓存方式的优化，大表数据分页的优化，数据检测模式的优化，可视化工具功能的拓展与完善等。 期待后续同事的改善升级。^\_^…….