**Nutch项目部分文件解释**

1. rootDir/conf/gora.properties

规定gora使用数据库类型的文件，hbase不需要用户名与密码

1. rootDir/conf/nutch-site.xml

nutch的配置文件

**storage.data.store.class**属性告知nutch使用的数据库类型

**http.agent.name**属性表示爬虫在爬取时的名字，一定要有，且在爬取淘宝网站时不能与百度爬虫同名，淘宝网的robots.txt拒绝了百度的爬虫爬取。

**plugin.folders** 属性指示nutch插件所在的位置，使用了绝对路径，记得改路径。

**http.content.limit** 属性表示爬取一个网站内容大小限制，遇到网站内容不能被完全爬取时可适当调大，默认65535。

还有很多其他属性，在遇到爬取过程有问题，又不报错的时候可以查看nutch-default.xml其中包含了能配置的属性以及英文解释，文件中的配置也是默认配置，nutch-site.xml的配置会覆盖nutch-default.xml的。

1. rootDir/conf/regex-urlfilter.txt

通过正则表达式过滤爬取网站网址的文件

1. rootDir/urls/seed.txt

储存最先被爬取网站的url

1. rootDir/docs/api

为nutch的英文api

1. InjectorJob.java

将seed.txt中的网址按照regex-normalize.xml和regex-urlfilter.txt过滤后添加至hbase数据库中

1. GeneratorJob.java

从数据库中取出所有未爬取的网址生成爬取列表

1. FetcherJob.java

按照爬取列表取得网址的页面内容，并存储在webpage表中

1. ParserJob.java

取出数据库中所有未解析的网址页面内容，解析并存储。

为了按照自己的需求找出内容需要对源码进行修改，通过阅读源码，找到需要修改的地方在rootDir\src\plugins\parse-html\src\java\org\apache\nutch\parse\html下

HtmlParser的getParse(String url, WebPage page)函数中可以得到当前解析网址的页面内容以及网址的有关信息例如url，然后按照自己需要的样子找到需要的信息，可以使用getParse中调用的nekohtml 的DOMtree遍历html节点的方法

1. DbupdatorJob.java

更新数据库信息

1. Mycrawler

可以运行Mycrawler.java来完成一次完整的爬取过程，main函数开头的设定了个别参数，常用的有

* **-dir** *dir* 指定用于存放抓取文件的目录名称。为inject的参数
* **-threads** *threads* 决定将会在获取是并行的线程数。为fetch的参数
* **-depth** *depth* 表明从根网页开始那应该被抓取的链接深度。为后四个job执行次数
* **-topN** *N* 决定在每一深度将会被取回的网页的最大数目。为generate的参数

通常情况下需要按照InjectorJob.java(只执行一次)----

GeneratorJob.java----FetcherJob.java----ParserJob.java----DbupdatorJob.java(执行-depth次，即深度，通过-depth次跳转可到达指定网址)顺序手动运行才可以完成爬取

1. 针对聚划算商品设置爬取间隔

经过判断，为聚划算明日预告的间隔设置为1天，同时修改FetchTime；聚划算商品详情网页爬取间隔为1天；第二次爬取相同的聚划算商品详情网页时，获取商品参与聚划算促销活动时间，并修改商品原详情页面爬取间隔，同时聚划算商品详情页面设为不再爬取；聚划算活动结束后第二次爬取商品原详情页面，同样设置为不再爬取。需要将以下类似的代码添加到FetcherJob的流程中可以获取到爬取对象Baseurl的源码位置

if( "forcast".equals(urlType))

setFetchInterval(1,"com.taobao.ju:http/tg/forcast.htm");

else if("detail.ju".equals(urlType)){

rowkey = "com.taobao.ju.detail:http/home.htm?id="+type\_id+ "&item\_id=" +item\_id;

if(Myhbase.QueryByCondition("webpage",rowkey).isEmpty())

//说明之前未爬取过

setFetchInterval(1,rowkey);

else setFetchInterval(-1,rowkey);

}

else {

if("taobao".equals(urlType))

rowkey = "com."+urltype+".item:http/item.html?id="+item\_id;

else

rowkey = "com."+urltype+".detail:http/item.html?id="+item\_id+"&tracelog=jubuybigpic";

if(Myhbase.QueryByCondition("webpage",rowkey).isEmpty())

setFetchInterval(duration,rowkey); //爬取间隔设置为商品参与聚划算的时间

else setFetchInterval(-1,rowkey);

}

void setFetchInterval(int days, rowkey)函数直接对hbase中的webpage表操作，将days乘以86400000(一天的毫秒数，即FetchTime的表示方式)，再加上Date类获取到的当前时间的毫秒数，赋值给webpage表中的FetchTime列的value，表示在这个时间之后执行爬虫会再次爬取这个网页。如果的到值为-1，则从webpage表中删除这一网页，表示爬取两次之后不再爬取该商品信息。

Result QueryByCondition(String tablename, String rowkey) (Myhabse.java中有类似函数)返回hbase根据表名和行名查询结果的Result对象，其中的方法isEmpty()为true时数据库中不存在查询对象。