1. 说一下爬虫程序执行的流程？（框架和三方库均可）？

2. 爬虫在向数据库存数据开始和结束都会发一条消息，是 scrapy 哪个模块实现的？

答：Item Pipeline scrapy 的信号处理使用的是 dispatch 模块

3. 爬取下来的数据如何去重，说一下具体的算法依据？

答：1.通过 MD5 生成电子指纹来判断页面是否改变

2.nutch 去重。nutch 中 digest 是对采集的每一个网页内容的 32 位哈希值，如果两个网页内容完全一样，它们的 digest 值肯定会一样。

4. 写爬虫是用多进程好？还是多线程好？ 为什么？

答：IO 密集型代码(文件处理、网络爬虫等)，多线程能够有效提升效率(单线程下有 IO 操作会进行 IO 等待，造成不必要的时间浪费，而开启多线程能在线程 A 等待时，自动切换到线程 B，可以不浪费 CPU 的资源，从而能提升程序执行效率)。在实际的数据采集过程中，既考虑网速和响应的问题，也需要考虑自身机器的硬件情况，来设置多进程或多线程。

5. 说一下 numpy 和 pandas 的区别？分别的应用场景？

答：Numpy 是　　的扩展包，纯数学。

Pandas 做　　以矩阵为基础的数学计算模块。提供了一套名为 DataFrame 的数据结构，比较契合统计分析中的表结构，并且提供了计算接口，可用 Numpy 或其它方式进行计算。

6. 验证码如何处理

1、Scrapy 自带处理验证码

2、获取到验证码图片的 url，调用第三方付费借口破解验证码

7. 微信公众号数据如何抓取？

sogou 微信搜索数据

8. 动态的股票信息如何抓取

股票数据的获取目前有如下两种方法可以获取:

1.http/JavaScript 接口取数据

2.web-service 接口

Sina 股票数据接口

以大秦铁路（股票代码：601006）为例，如果要获取它的最新行情，只需访问新浪的股票数据

接口：http://hq.sinajs.cn/list=sh601006 这个 url 会返回一串文本，例如 var hq_str_sh601006="大秦铁路, 27.55, 27.25, 26.91, 27.55, 26.20, 26.91, 26.92,

22114263, 589824680, 4695, 26.91, 57590, 26.90, 14700, 26.89, 14300,

26.88, 15100, 26.87, 3100, 26.92, 8900, 26.93, 14230, 26.94, 25150, 26.95, 15220, 26.96, 2008-01-11, 15:05:32";

9. 爬虫部署

Scrapyd

http://blog.csdn.net/xiaoquantouer/article/details/53164306

10. scrapy 去重

数据量不大时，可以直接放在内存里面进行去重，python 可以使用 set()进行去重。

当去重数据需要持久化时可以使用 redis 的 set 数据结构。

当数据量再大一点时,可以用不同的加密算法先将长字符串压缩成 16/32/40 个字符，再使用上面两种方法去重;

当数据量达到亿（甚至十亿、百亿）数量级时，内存有限，必须用“位”来去重，才能够满足需求。Bloomfilter 就是将去重对象映射到几个内存“位”，通过几个位的 0/1 值来判断一个对象是否已经存在。

然而 Bloomfilter 运行在一台机器的内存上，不方便持久化（机器 down 掉就什么都没啦），也不方便分布式爬虫的统一去重。如果

可以在 Redis 上申请内存进行 Bloomfilter，以上两个问题就都能解决了。

simhash 最牛逼的一点就是将一个文档，最后转换成一个 64 位的字节，暂且称之为特征字，然后判断重复只需要判断他们的特征字的距离是不是<n（根据经验这个 n 一般取值为 3），就可以判断两个文档是否相似。

11. 分布式有哪些方案，哪一种最好？

celery、beanstalk，gearman

个人认为 gearman 比较好。原因主要有以下几点：

1）.技术类型简单，维护成本低。

2）.简单至上。能满足当前的技术需求即可（分布式任务处理、异步同步任务同时支持、任务队列的持久化、维护部署简单）。

3）.有成熟的使用案例。instagram 就是使用的 gearman 来完成图片的处理的相关任务，有成功的经验，我们当然应该借鉴。

12. Post 和 get 区别

1、GET 请求，请求的数据会附加在 URL 之后，以?分割 URL 和传输数据，多个参数用&连接。URL 的编码格式采用的是 ASCII 编码，而不是 uniclde，即是说所有的非 ASCII 字符都要编码之后再传输。

POST 请求:POST 请求会把请求的数据放置在 HTTP 请求包的包体中。上面的 item=bandsaw 就是实际的传输数据。

因此，GET 请求的数据会暴露在地址栏中，而 POST 请求则不会。

2、传输数据的大小

在 HTTP 规范中，没有对 URL 的长度和传输的数据大小进行限制。但是在实际开发过程中，对于 GET，特定的浏览器和服务器对 URL 的长度有限制。因此，在使用 GET 请求时，传输数据会受到 URL 长度的限制。

对于 POST，由于不是 URL 传值，理论上是不会受限制的，但是实际上各个服务器会规定对 POST 提交数据大小进行限制，Apache、IIS 都有各自的配置。

3、安全性

POST 的安全性比 GET 的高。这里的安全是指真正的安全，而不同于上面 GET 提到的安全方法中的安全，上面提到的安全仅仅是不修改服务器的数据。比如，在进行登录操作，通过 GET 请求，用户名和密码都会暴露再 URL 上，因为登录页面有可能被浏览器缓存以及其他人查看浏览器的历史记录的原因，此时的用户名和密码就很容易被他人拿到了。除此之外，GET 请求提交的数据还可能会造成 Cross-site request frogery 攻击。

13. 为什么要三次握手和四次挥手？

建立连接的过程是利用客户服务器模式，假设主机 A 为客户端，主机 B 为服务器端。

（1）TCP 的三次握手过程：主机 A 向 B 发送连接请求；主机 B 对收到的主机 A 的报文段进行确认；主机 A 再次对主机 B 的确认进行确认。

（2）采用三次握手是为了防止失效的连接请求报文段突然又传送到主机 B，因而产生错误。失效的连接请求报文段是指：主机 A 发出的连接请求没有收到主机 B 的确认，于是经过一段时间后，主机 A 又重新向主机 B 发送连接请求，且建立成功，顺序完成数据传输。考虑这样一种特殊情况，主机 A 第一次发送的连接请求并没有丢失，而是因为网络节点导致延迟达到主机 B，主机 B 以为是主机 A 又发起的新连接，于是主机 B 同意连接，并向主机 A 发回确认，但是此时主机 A 根本不会理会，主机 B 就一直在等待主机 A 发送数据，导致主机 B 的资源浪费。

（3）采用两次握手不行，原因就是上面说的失效的连接请求的特殊情况，因此采用三次握手刚刚好，两次可能出现失效，四次甚至更多次则没必要，反而复杂了

四次挥手：

先由客户端向服务器端发送一个 FIN，请求关闭数据传输。

当服务器接收到客户端的 FIN 时，向客户端发送一个 ACK，其中 ack 的值等于 FIN+SEQ

然后服务器向客户端发送一个 FIN，告诉客户端应用程序关闭。

当客户端收到服务器端的 FIN 是，回复一个 ACK 给服务器端。其中 ack 的值等于 FIN+SEQ

为什么要 4 次挥手？

确保数据能够完成传输

14. 多线程有哪些模块？

在 Python 中可使用的多线程模块主要有两个，thread 和 threading 模块。Queue。

15. 谈一谈你对 Selenium 和 PhantomJS 了解

Selenium 是一个 Web 的自动化测试工具，可以根据我们的指令，让浏览器自动加载页面，获取需要的数据，甚至页面截屏，或者判断网站上某些动作是否发生。Selenium 自己不带浏览器，不支持浏览器的功能，它需要与第三方浏览器结合在一起才能使用。但是我们有时候需要让它内嵌在代码中运行，所以我们可以用一个叫 PhantomJS 的工具代替真实的浏览器。Selenium 库里有个叫 WebDriver 的 API。WebDriver 有点儿像可以加载网站的浏览器，但是它也可以像 BeautifulSoup 或者其他 Selector 对象一样用来查找页面元素，与页面上的元素进行交互（发送文本、点击等），以及执行其他动作来运行网络爬虫。

PhantomJS 是一个基于 Webkit 的“无界面”(headless)浏览器，它会把网站加载到内存并执行页面上的 JavaScript，因为不会展示图形界面，所以运行起来比完整的浏览器要高效。

如果我们把 Selenium 和 PhantomJS 结合在一起，就可以运行一个非常强大的网络爬虫了，这个爬虫可以处理 JavaScrip、Cookie、headers，以及任何我们真实用户需要做的事情。

16. 常见的反爬虫和应对方法？

1）.通过 Headers 反爬虫

从用户请求的 Headers 反爬虫是最常见的反爬虫策略。很多网站都会对 Headers 的 User-Agent 进行检测，还有一部分网站会对 Referer 进行检测（一些资源网站的防盗链就是检测 Referer）。如果遇到了这类反爬虫机制，可以直接在爬虫中添加 Headers，将浏览器的 User-Agent 复制到爬虫的 Headers 中；或者将 Referer 值修改为目

标网站域名。对于检测 Headers 的反爬虫，在爬虫中修改或者添加 Headers 就能很好的绕过。

2）.基于用户行为反爬虫

还有一部分网站是通过检测用户行为，例如同一 IP 短时间内多次访问同一页面，或者同一账户短时间内多次进行相同操作。

大多数网站都是前一种情况，对于这种情况，使用 IP 代理就可以解决。可以专门写一个爬虫，爬取网上公开的代理 ip，检测后全部保存起来。这样的代理 ip 爬虫经常会用到，最好自己准备一个。有了大量代理 ip 后可以每请求几次更换一个 ip，这在 requests 或者 urllib2 中很容易做到，这样就能很容易的绕过第一种反爬虫。

对于第二种情况，可以在每次请求后随机间隔几秒再进行下一次请求。有些有逻辑漏洞的网站，可以通过请求几次，退出登录，重新登录，继续请求来绕过同一账号短时间内不能多次进行相同请求的限制。

3）.动态页面的反爬虫

上述的几种情况大多都是出现在静态页面，还有一部分网站，我们需要爬取的数据是通过 ajax 请求得到，或者通过 JavaScript 生成的。首先用 Fiddler 对网络请求进行分析。如果能够找到 ajax 请求，也能分析出具体的参数和响应的具体含义，我们就能采用上面的方法，直接利用 requests 或者 urllib2 模拟 ajax 请求，对响应的 json 进行分析得到需要的数据。

能够直接模拟 ajax 请求获取数据固然是极好的，但是有些网站把 ajax 请求的所有参数全部加密了。我们根本没办法构造自己所需要的数据的请求。这种情况下就用 selenium+phantomJS，调用浏览器内核，并利用 phantomJS 执行 js 来模拟人为操作以及触发页面中的 js 脚本。从填写表单到点击按钮再到滚动页面，全部都可以模拟，不考虑具体的请求和响应过程，只是完完整整的把人浏览页面获取数据的过程模拟一遍。

用这套框架几乎能绕过大多数的反爬虫，因为它不是在伪装成浏览器来获取数据（上述的通过添加 Headers 一定程度上就是为了伪装成浏览器），它本身就是浏览器，phantomJS 就是一个没有界面的浏览器，只是操控这个浏览器的不是人。利 selenium+phantomJS 能干很多事情，例如识别点触式（12306）或者滑动式的验证码，对页面表单进行暴力破解等。

17. 动态加载又对及时性要求很高怎么处理

Selenium+Phantomjs

尽量不使用 sleep 而使用 WebDriverWait

18. 分布式爬虫主要解决什么问题？

1)ip

2)带宽

3）cpu

4）io

19. 什么是 URL？

URL，即统一资源定位符，也就是我们说的网址，统一资源定位符是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的 URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它。

20. python 爬虫有哪些常用技术？

Scrapy，Beautiful Soup，urllib，urllib2，requests

21. 简单说一下你对 scrapy 的了解？

scrapy 是一个快速(fast)、高层次(high-level)的基于 python 的 web 爬虫构架。用来下载、并解析 web 页面，其 parse->yield item->pipeline 流程是所有爬虫的固有模式。构造形式主要分

spider.pypipeline.py item.py decorator.py middlewares.py setting.py。

22. Scrapy 的优缺点?

优点：scrapy 是异步的

采取可读性更强的 xpath 代替正则

强大的统计和 log 系统

同时在不同的 url 上爬行

支持 shell 方式，方便独立调试

写 middleware,方便写一些统一的过滤器

通过管道的方式存入数据库

缺点：基于 python 的爬虫框架，扩展性比较差

基于 twisted 框架，运行中的 exception 是不会干掉 reactor，并且异步框架出错后是不会停掉其他任务的，数据出错后难以察觉。

23. scrapy 和 request?

scrapy 是封装起来的框架，他包含了下载器，解析器，日志及异常处理，基于多线程， twisted 的方式处理，对于固定单个网站的爬取开发，有优势，但是对于多网站爬取 100 个网站，并发及分布式处理方面，不够灵活，不便调整与括展。

request 是一个 HTTP 库， 它只是用来，进行请求，对于 HTTP 请求，他是一个强大的库，下载，解析全部自己处理，灵活性更高，高并发与分布式部署也非常灵活，对于功能可以更好实现.

24. 网络传输层？

应用层—http ftp dns nfs

传输层—tcp --udp

网络层—ip icmp igmp

链路层—data link

物理层—media

25. 设置 ip 和掩码

Ifconfig eth0 192.168.13.225 netmask 255.255.255.0

26. 设置网关

route add default gw 192.168.5.1

27. 什么是 2MSL?

2MSL 即两倍的 MSL，TCP 的 TIME_WAIT 状态也称为 2MSL 等待状态，当 TCP 的一端发起主动关闭，在发出最后一个 ACK 包后，即第 3 次握手完成后发送了第四次握手的 ACK 包后就进入了 TIME_WAIT 状态，必须在此状态上停留两倍的 MSL 时间，等待 2MSL 时间

主要目的是怕最后一个 ACK 包对方没收到，那么对方在超时后将重发第三次握手的 FIN 包，主动关闭端接到重发的 FIN 包后可以再发一个 ACK 应答包。在 TIME_WAIT 状态时两端的端口不能使用，要等到 2MSL 时间结束才可继续使用。当连接处于 2MSL 等待阶段时任何迟到的报文段都将被丢弃。不过在实际应用中可以通过设置 SO_REUSEADDR 选项达到不必等待 2MSL 时间结束再使用此端口。

28. 创建一个简单 tcp 服务器需要的流程？

1.socket 创建一个套接字

2.bind 绑定 ip 和 port

3.listen 使套接字变为可以被动链接

4.accept 等待客户端的链接

5.recv/send 接收发送数据

29. TTL，MSL，RTT？

MSL：报文最大生存时间”，他是任何报文在网络上存在的最长时间，超过这个时间报文将被丢弃。

TTL：TTL 是 time to live 的缩写，中文可以译为“生存时间”，这个生存时间是由源主机设置初始值但不是存的具体时间，而是存储了一个 ip 数据报可以经过的最大路由数，每经过一个处理他的路由器此值就减 1，当此值为 0 则数据报将被丢弃，同时发送 ICMP 报文通知源主机。RFC 793 中规定 MSL 为 2 分钟，实际应用中常用的是 30 秒，1 分钟和 2 分钟等。TTL 与 MSL 是有关系的但不是简单的相等的关系，MSL 要大于等于 TTL。

RTT： RTT 是客户到服务器往返所花时间（round-trip time，简称 RTT），TCP 含有动态估算 RTT 的算法。TCP 还持续估算一个给定连接的 RTT，这是因为 RTT 受网络传输拥塞程序的变化而变化。

30. 常用的反爬虫措施？

1.添加代理

2.降低访问频率

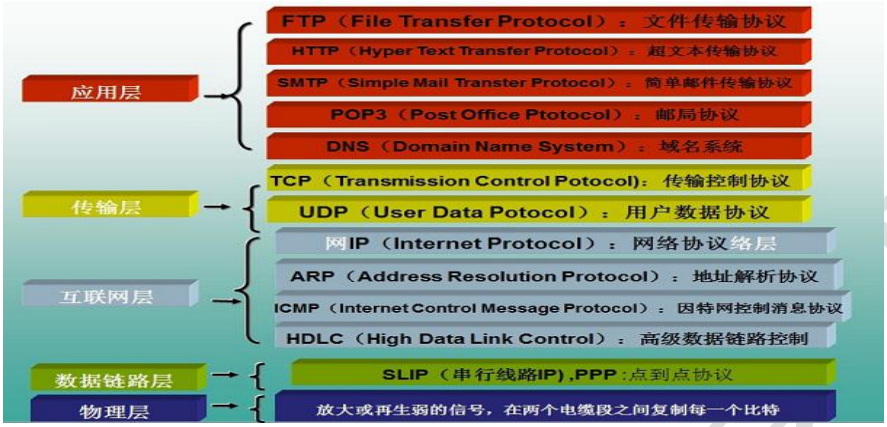
3. User-Agent

4. **动态 HTML 数据加载**

5. 验证码处理

6. Cookie

31. 网络协议概述



32. 关于 HTTP/HTTPS 的区别，分别应该在什么场合下。

HTTPS 和 HTTP 的区别：

https 协议需要到 ca 申请证书，一般免费证书很少，需要交费。

http 是超文本传输协议，信息是明文传输，https 则是具有安全性的 ssl 加密传输协议

http 和 https 使用的是完全不同的连接方式用的端口也不一样,前者是 80,后者是 443。

http 的连接很简单,是无状态的

HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议 要比 http 协议安全

应用场合：

http:适合于对传输速度，安全性要求不是很高，且需要快速开发的应用。如 web 应用，小的手机游戏等等.

https:https 应该用于任何场景！

43. HTTPS 有什么优点和缺点

优点：1、使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；

2、HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全，可防止数据在传输过程中不被窃取、改变，确保数据的完整性。

3、HTTPS 是现行架构下最安全的解决方案，虽然不是绝对安全，但它大幅增加了中间人攻击的成本

缺点：

1.HTTPS 协议的加密范围也比较有限，在黑客攻击、拒绝服务攻击、服务器劫持等方面几乎起不到什么作用

2.HTTPS 协议还会影响缓存，增加数据开销和功耗，甚至已有安全措施也会受到影响也会因此而受到影响。

3.SSL 证书需要钱。功能越强大的证书费用越高。个人网站、小网站没有必要一般不会用。

4.HTTPS 连接服务器端资源占用高很多，握手阶段比较费时对网站的相应速度有负面影响。

5.HTTPS 连接缓存不如 HTTP 高效。

44. HTTPS 是如何实现安全传输数据的。

HTTPS 其实就是在 HTTP 跟 TCP 中间加多了一层加密层 TLS/SSL。SSL 是个加密套件，负责对 HTTP 的数据进行加密。TLS 是 SSL 的升级版。现在提到 HTTPS，加密套件基本指的是 TLS。原先是应用层将数据直接给到 TCP 进行传输，现在改成应用层将数据给到 TLS/SSL，将数据加密后，再给到 TCP 进行传输。

33. HTTPS 安全证书是怎么来的，如何申请，国内和国外有哪些第三方机构提供安全证书认证。

国内：沃通(WoSign)

中国人民银行联合 12 家银行建立的金融 CFCA

中国电信认证中心（CTCA）

海关认证中心（SCCA）

国家外贸部 EDI 中心建立的国富安 CA 安全认证中心

SHECA（上海 CA）为首的 UCA 协卡认证体系

国外：StartSSL

GlobalSign

GoDaddy

Symantec

34. get 和 post 请求有什么区别，分别应该在什么场合下。

区别：

get:从指定的服务器中获取数据。

GET 请求能够被缓存

GET 请求会保存在浏览器的浏览记录中

以 GET 请求的 URL 能够保存为浏览器书签

GET 请求有长度限制

GET 请求主要用以获取数据

post:

POST 请求不能被缓存下来

POST 请求不会保存在浏览器浏览记录中

以 POST 请求的 URL 无法保存为浏览器书签

POST 请求没有长度限制

POST 请求会把请求的数据放置在 HTTP 请求包的包体中,POST 的安全性比 GET 的高.可能修改变服务器上的资源的请求.

应用场合：

post：请求的结果有持续性的副作用（数据库内添加新的数据行）

若使用 GET 方法，则表单上收集的数据可能让 URL 过长。

要传送的数据不是采用 7 位的 ASCII 编码。

get：请求是为了查找资源，HTML 表单数据仅用来帮助搜索。

请求结果无持续性的副作用。

收集的数据及 HTML 表单内的输入字段名称的总长不超过 1024 个字符

35. HTTP 请求会有哪些信息发送到后台服务器。

请求行 （请求方式、资源路径和 HTTP 协议版本）POST /demo/login HTTP/1.1

请求消息头

消息正文（也叫实体内容） username=xxxx&password=1234

36. 描述下 scrapy 框架运行的机制？

答：从 start_urls 里获取第一批 url 并发送请求，请求由引擎交给调度器入请求队列，获取完毕后，调度器将请求队列里的请求交给下载器去获取请求对应的响应资源，并将响应交给自己编写的解析方法做提取处理：1. 如果提取出需要的数据，则交给管道文件处理；2. 如果提取出 url，则继续执行之前的步骤（发送 url 请求，并由引擎将请求交给调度器入队列...），直到请求队列里没有请求，程序结束。

37. scrapy 和 scrapy-redis 有什么区别？为什么选择 redis 数据库？

1) scrapy 是一个 Python 爬虫框架，爬取效率极高，具有高度定制性，但是不支持分布式。而 scrapy-redis 一套基于 redis 数据库、运行在 scrapy 框架之上的组件，可以让 scrapy 支持分布式策略，Slaver 端共享 Master 端 redis 数据库里的 item 队列、请求队列和请求指纹集合。

2) 为什么选择 redis 数据库，因为 redis 支持主从同步，而且数据都是缓存在内存中的，所以基于 redis 的分布式爬虫，对请求和数据的高频读取效率非常高。

38. 实现模拟登录的方式有哪些？

1) 使用一个具有登录状态的 cookie，结合请求报头一起发送，可以直接发送 get 请求，访问登录后才能访问的页面。

2) 先发送登录界面的 get 请求，在登录页面 HTML 里获取登录需要的数据（如果需要的话），然后结合账户密码，再发送 post 请求，即可登录成功。然后根据获取的 cookie 信息，继续访问之后的页面。

39. 简单介绍下 scrapy 的异步处理。

scrapy 框架的异步机制是基于 twisted 异步网络框架处理的，在 settings.py 文件里可以设置具体的并发量数值（默认是并发量 16）。