北京交通大学
BEIJING JIAOTONG UNIVERSITY

《大数据概论》
大数据感知与获取

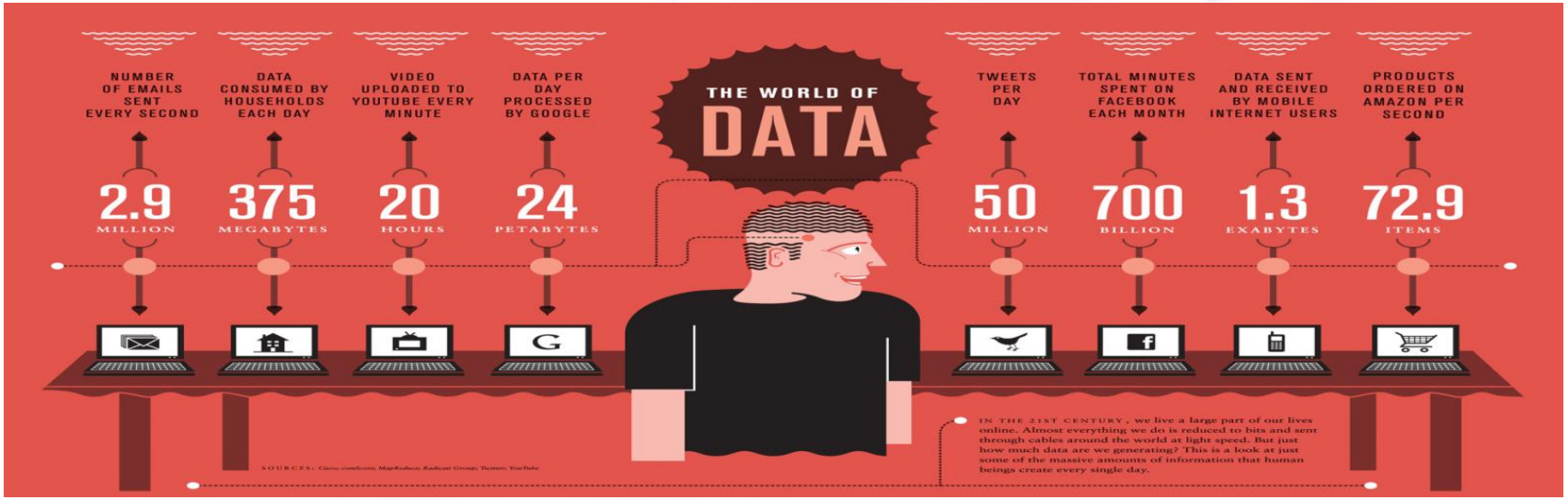
鲍鹏
软件学院

大数据感知与获取

- 大数据渠道
- 大数据获取
  - 内部数据及其获取方法
  - 外部数据及其获取方法

大数据感知的时代背景

半个世纪以来，随着计算机技术全面融入社会生活，信息爆炸已经积累到了一个开始引发变革的程度。它不仅使世界充斥着比以往更多的信息，而且其增长速度也在加快。信息爆炸的学科如天文学和基因学，创造出了“大数据”这个概念。如今，这个概念几乎应用到了所有人类智力与发展的领域中。



20世纪90年代，数据仓库之父的Bill Inmon就经常提及Big Data。

大数据感知渠道–来源

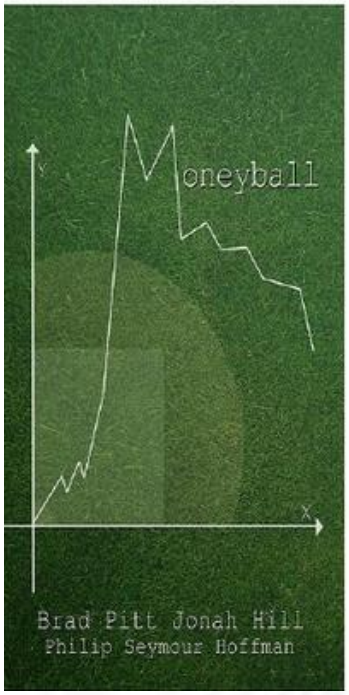
21世纪是数据信息大发展的时代，移动互联、社交网络、电子商务等极大拓展了互联网的边界和应用范围，各种数据正在迅速膨胀并变大。

互联网（社交、搜索、电商）、
移动互联网（微博）、
物联网（传感器，智慧地球）、
车联网、GPS、医学影像、安全监控、
金融（银行、股市、保险）、
电信（通话、短信）**都在疯狂产生着数据。**

2011年5月，在“云计算相遇大数据”为主题的EMC World 2011 会议中，EMC 抛出了Big Data概念。

大数据感知渠道-决策

布拉德•皮特主演的《点球成金》是一部美国奥斯卡获奖影片，所讲述的是皮特扮演的棒球队总经理利用计算机数据分析，对球队进行了翻天覆地的改造，让一家不起眼的小球队能够取得巨大的成功。



基于历史数据，利用数据建模定量分析不同球员特点，合理搭配，重新组队；

打破传统思维，通过分析比赛数据，寻找“性价比”最高球员，运用数据取得成功；

数据感知->辅助决策

大数据感知渠道-价值

仅供开采162年

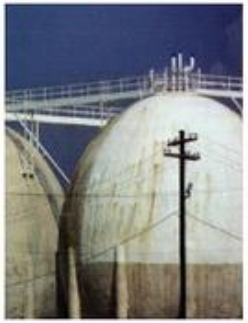
数据不再是社会生产的“副产物”，而是可被二次乃至多次加工的原料，从中可以探索更大价值，它变成了生产资料。

过去3年数据总量被以往4万年还多

2013年,10分钟的信息总量将达1.8ZB

2010年全球数据总量1.2ZB，年增长50%

仅供开采45年



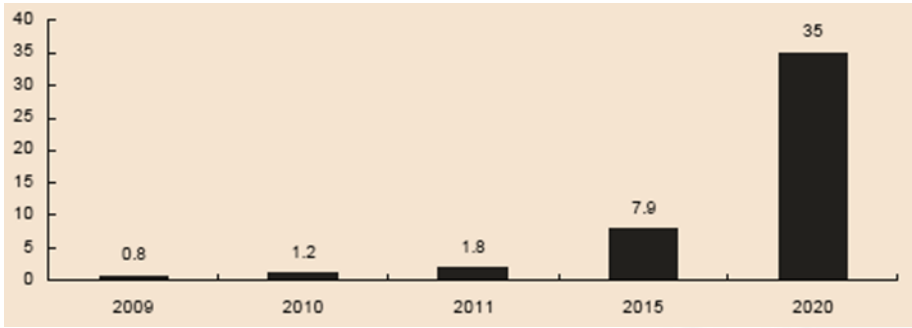
仅供开采60年

不可再生资源VS数据

数据被认为是信息时代的基础生活资料与市场要素
重要程度不亚于物资资产和人力资本

大数据感知渠道-挑战

数据量增加

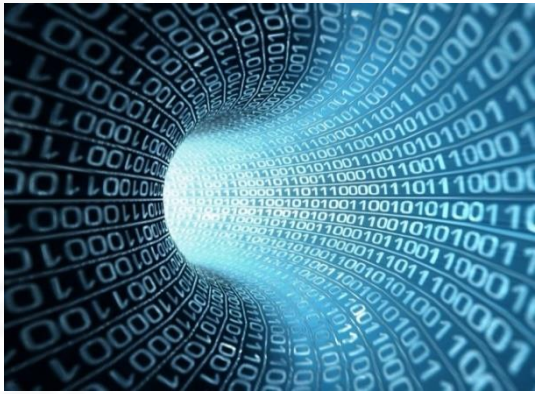


根据IDC 监测，人类产生的数据量正在呈指数级增长，大约每两年翻一番，这个速度在2020 年之前会继续保持下去。这意味着人类在最近两年产生的数据量相当于之前产生的全部数据量。

TB ⇨ PB ⇨ EB ⇨ ZB

数据结构日趋复杂

大量新数据源的出现则导致了非结构化、半结构化数据爆发式的增长



- 这些由我们创造的信息背后产生的这些数据早已经远远超越了目前人力所能处理的范畴

大数据感知渠道-挑战

1. Volume

数据量巨大

全球在2010 年正式进入ZB 时代，IDC预计到2020 年，全球将总共拥有35ZB 的数据量

2. Variety

结构化数据、半结构化数据和非结构化数据

如今的数据类型早已不是单一的文本形式，订单、日志、音频，多类型的数据对数据处理能力提出了更高的要求

3. value

沙里淘金，价值密度低

以视频为例，一部一小时的视频，在连续不间断监控过程中，可能有用的数据仅仅只有一两秒。如何通过强大的机器算法更迅速地完成数据的价值“提纯”是目前大数据汹涌背景下亟待解决的难题

4. Velocity

实时获取需要的信息

大数据区分于传统数据最显著的特征。如今已是ZB时代，在如此海量的数据面前，处理数据的效率就是企业的生命

大数据感知渠道-挑战

	传统数据	大数据
数据产生方式	被动采集数据	主动生成数据
数据采集密度	采样密度较低，采样数据有限	利用大数据平台，可对需要分析事件的数据进行密度采样，精确获取事件全局数据
数据源	数据源获取较为孤立，不同数据之间添加的数据整合难度较大	利用大数据技术，通过分布式技术、分布式文件系统、分布式数据库等技术对多个数据源获取的数据进行整合处理
数据处理方式	大多采用离线处理方式，对生成的数据集中分析处理，不对实时产生的数据进行分析	较大的数据源、响应时间要求低的应用可以采取批处理方式集中计算；响应时间要求高的实时数据处理采用流处理的方式进行实时计算，并通过对历史数据的分析进行预测分析

大数据感知与获取

- 大数据渠道
- 大数据获取
  - 内部数据及其获取方法
  - 互联网数据及其获取方法

大数据获取

内部数据	不同的利益主体（包括政府各个部门、企事业单位等）出于自身职能定位和获益诉求而建设的IT系统在完成本部门既定角色目标任务过程中，有意或者无意地存储下有关物理世界实体对象的各类数据。
互联网数据	通过不同的互联网应用产品而沉淀在互联网中的各类数据。存放在不同利益主体的服务器中，基于开放、共享精神，人人都可以通过浏览网页（或者通过APP）的形式访问这些数据。

大数据感知与获取

- 大数据渠道
- 大数据获取
  - 内部数据及其获取方法
  - 互联网数据及其获取方法

内部数据获取方法

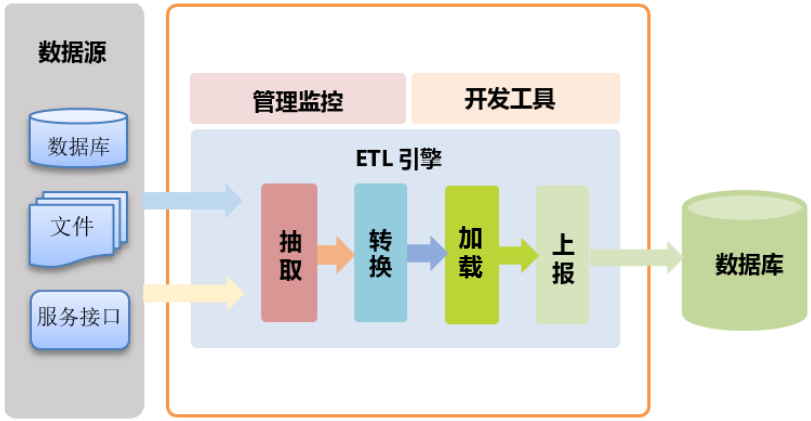
对于一个企业来说，企业数据不仅包括本企业自己生产的数据也有其他企业合作时可以获得的数据。面对大数据时代带来的机遇和挑战，内部数据资源整合是现代企业必须具备的能力和强有力的竞争优势，具体体现在如下几个方面：

- 构建数据驱动应用，推进拓展价值实现
- 统一数据规范标准，推动数据共享开放
- 重视数据安全管理，完善数据安全保障
- 推进数据融合管理，增加数据语义厚度

内部数据获取方法

不同用户和企业内部不同部门提供的内部数据可能来自不同的途径，其数据内容、数据格式和数据质量千差万别，因此，**能否对数据进行有效的整合将成为是否能够对内部数据进行有效利用和关键，ETL是其中重要的处理手段：**

ETL(Extract-Transform-Load)是数据的抽取、转换、装载的过程，负责完成数据从数据源向目标数据仓库的转化。即用户从数据源抽取所需的数据，经过数据清洗，按照预先定义的数据仓库模型，最终将数据加载入数据仓库。



ETL-抽取

数据抽取是从数据源中抽取数据的过程，由于数据会存放在数据库里，数据抽取也就变成了从数据库中抽取数据的过程。由于大多数场景下，数据会存放在数据库里，从数据库中抽取数据一般分为两种方式：

1）全量抽取

全量抽取就是对整个数据库的所有数据进行抽取，将数据源库中的所有数据原封不动的从数据库中抽取出来，然后转换成ETL工具可以识别的格式。

2）增量抽取

增量抽取只抽取自上次抽取以来数据库中新增或修改的数据。优秀的捕获方法应该做到能够将数据库中的变化数据以较高的准确率获得的同时不对业务系统造成太大的压力而影响现有业务。

ETL-抽取

在**增量数据**抽取过程中，常用的捕获变化数据的方法有：日志对比、时间戳、触发器、全表比对等。

1）**日志比对：**通过分析数据库自身的日志来判断变化的数据。

2）**时间戳：**通过增加一个时间戳字段，在更新修改表数据的同时修改时间戳的值。当进行数据抽取时，通过比较系统时间与实践戳字段的值来决定抽取哪些数据。

3）**触发器 ：**在数据源表上建立触发器，例如可以建立插入、修改、删除三个触发器，每当源表中的数据发生变化，就通过相应的护法其将变化的数据写入一个临时表，抽取线程从临时表中抽取数据，临时表中抽取过的数据被标记或删除。

ETL-转换

从数据源中抽取的数据不一定满足目的数据库的要求，需要对抽取出的数据进行数据转换，主要有两种操作方式。

1）**ETL引擎中的数据转换和加工：**ETL引擎中一般以组件化的方式实现数据转换，常用的数据转换组件有字段映射、数据过滤、数据替换、数据计算、数据验证、数据加解密、数据合并、数据拆分等。

2）**在数据库中进行数据加工：**关系数据库本身已经提供强大的SQL指令、函数来支持数据的加工，如在SQL查询语句中添加where条件进行过滤、查询中重复名字段名与目的表进行映射等。

ETL-转换

数据转换把已抽取的数据升华为数据仓库的有效数据，通过设计转换规则，实施过滤、合并、解码和翻译等操作完成。数据转换需要理解业务侧重、信息需求和可用源数据，常用规则如下：

1）**字段级的转换。**主要是指数据类型转换，增加“上下文”数据，例如时间戳；将数值型的地域编码替换成地域名称，如解码(decoding)等。

2）**清洁和净化。**主要是保留字段具有特定值或特定范围的记录；引用完整性检查；去除重复记录等。

3）**多数据源整合。**字段映射(mapping)、代码变换(transposing)、合并(merging) 、派生(derivation)。

4）**聚合(aggregation)和汇总(summarization)。**事务性数据库侧重于细节，数据仓库侧重于高层次的聚合和汇总。

数据清洗

数据清洗是指在数据集中发现不准确、不完整或不合理数据，并对这些数据进行修补或移除以提高数据质量的过程，主要步骤包括：

1）定义错误类型

2）搜索并标识错误实例

3）改正错误

4）文档记录错误实例和错误类型

5）修改数据录入程序以减少未来的错误。

ETL-加载

将转换和加工后的数据加载到目的库中通常是ETL最后步骤，加载数据的最佳方式取决于所执行操作的类型以及需要装入多少数据，当目的库时关系数据库时，一般有两种装载方式：

1）直接中SQL语句进行插入、更新、删除操作。

2）采用批量装载方式，数据库特有的批量装载工具或者API.

ETL常用工具

- ETL数据整合主流的工具及其特点

比较项目		DataPipeline	Kettle	Oracle Goldengate	Informatica
设计及架构	适用场景	用于数据融合、数据交换场景，专为超大数据量、高度复杂数据链路设计的数据交换平台	面向数据仓库建模，传统ETL工具	主要用于数据备份、容灾	面向数据仓库建模，传统ETL工具
	使用方式	全流程图形化界面，应用端采用B/S架构，Cloud Native为云而生，所有操作在浏览器内就可以完成，不需要额外的开发和生产发布	C/S客户端模式，开发和生产环境需要独立部署，任务的编写、调试、修改都在本地，需要发布到生产环境。线上生产环境没有界面，需要通过日志来调试debug	没有图形化的界面，操作皆为命令行方式，可配置能力差	C/S客户端模式，开发和生产环境需要独立部署，任务的编写、调试、修改都在本地，需要发布到生产环境。学习成本较高，需要受过专业培训的工程师才能使用
	低层架构	分布式集群高可用架构，支持多节点扩展，支持超大数据量，架构容错性高。在节点之间自动调节任务分配，适用于大数据场景	主从结构属于非高可用架构，扩展性差，架构容错性低，不适用大数据场景	可做集群部署，规避单点故障，依赖于外部环境，如Oracle RAC等	schema mapping非自动；可复制性比较差；更新换代不是很强

ETL常用工具

- ETL数据整合主流的工具及其特点

比较项目		DataPipeline	Kettle	Oracle Goldengate	Informatica
功能	CDC（Change Data Capture，改变数据捕获）机制	基于日志、基于时间戳和自增序列等多种方式可选	基于时间戳、触发器等	主要是基于日志	基于日志、基于时间戳和自增序列等多种方式可选
	对数据库的影响	基于日志的采集方式对数据库无侵入性	对数据库表结构有要求，存在一定侵入性	源端数据库需要预留额外的缓存空间	基于日志的采集方式对数据库无侵入性
	自动断点续传	支持	不支持	支持	不支持
	监控预警	可视化的过程监控，提供多样化的图表，辅助运维，故障问题可实时预警	依赖日志定位故障问题，属于后处理的方式，缺少过程预警	无图形化的界面预警	可以看到报错信息，信息相对笼统，定位问题仍需依赖分析日志
	数据清洗	围绕数据质量做轻量清洗	根据数据仓库的数据需求来建模计算，清洗功能相对复杂，需要手动设置	轻量清洗	支持复杂逻辑的清洗和转化
	数据转换	自动化的schema mapping	手动配置schema mapping	需手动配置异构数据间的映射	手动配置schema mapping

ETL常用工具

- ETL数据整合主流的工具及其特点

比较项目		DataPipeline	Kettle	Oracle Goldengate	Informatica
特性	数据实时性	实时	非实时	实时	支持实时。主流应用基于时间戳等进行批量处理，实时效率未知
	应用难度	低	低	中	高
	是否需要开发	否	是	是	是
	易用性	高	高	中	低
	稳定性	高	高	高	中
其他	实施及售后	原厂实施和售后服务	开源软件，需自客户自行实施、维护	原厂和第三方的实施和售后服务	主要为第三方的实施和售后服务

ETL常用工具

- ETL数据整合主流的工具中，Kettle是业界比较受欢迎、使用人数较多且应用较广泛的ETL数据整合工具，深受用户的喜爱。

序号	优势	描述
1	开源软件，无需付费，技术支持强	纯Java编写，即使商业用户也没有限制。出现问题可以到社区咨询，技术支持遍布全世界
2	图形界面，易用性	有非常容易使用的GUI图形用户界面（Graphical User Interface，简称GUI），基本上无须培训
3	部署简单，无须安装	纯Java编写，支持多平台，无须安装
4	强大的基础数据转换和工作流控制	Transformation转换和Job作业两种脚本文件，强大的基础数据转换和工作流控制，有较好的监控日志
5	全面的数据访问和支持	支持非常广泛的数据库和数据文件，可以通过插件扩展
6	要求技能不高，上手容易	了解数据建模，熟悉ETL设计和SQL语句操作即可

大数据感知与获取

- 大数据渠道
- 大数据获取
  - 内部数据及其获取方法
  - 互联网数据及其获取方法

互联网数据

互联网技术的发展在改变人们生活的同时也产生了大量的网络数据，例如交易数据、博文图片信息等，互联网大数据通常是指“人、机、物”三元世界在网络空间中彼此之间相互交互与融合所产生的并在互联网上可以获得的大数据，其所具有的特性包括：

1）多源异构性

2）交互性

3）时效性

4）社会性

5）突发性

6）高噪音

互联网数据

多源异构性：网络大数据通常由不同的用户、不同的网站产生，数据形式也呈现出不同的形式，如语音、视频、图片和文本。

互联网数据

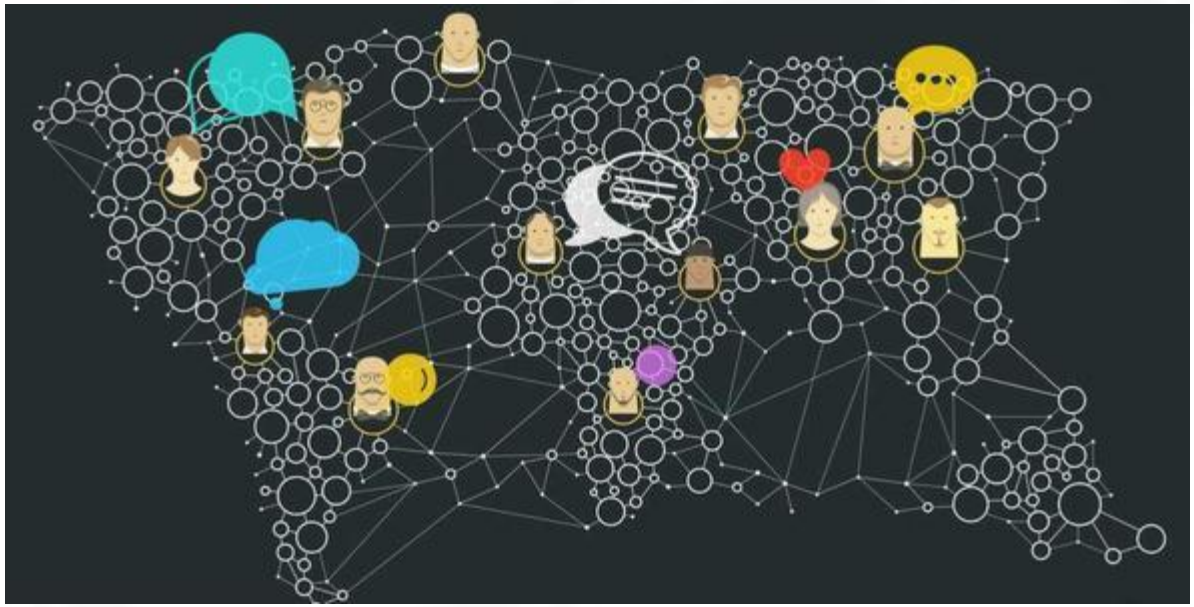
交互性：不同于测量和传感器获取的大规模科学数据（如气象数据、卫星遥感数据），微博、微信、Facebook、Twitter等社交网络兴起导致大量网络数据具有较强的交互性。

互联网数据

时效性：在互联网和移动互联网平台上，每时每刻都有大量的新数据发布，网络大数据内容不断变化，使得信息传播具有时序相关性。

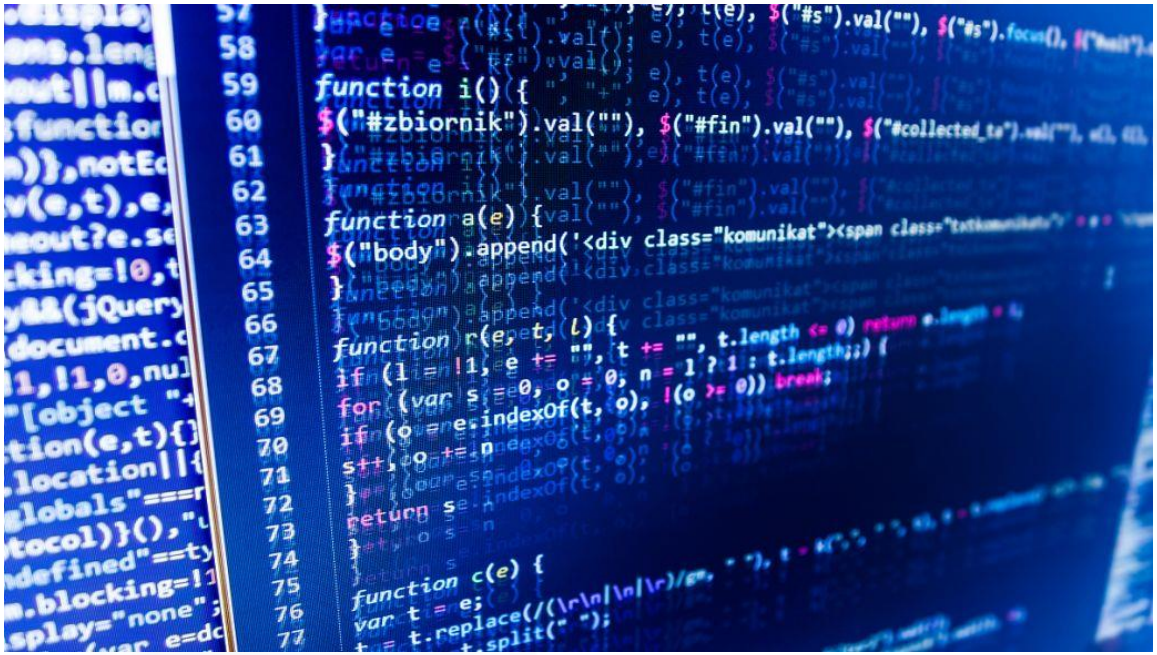
互联网数据

社会性：网络上用户不仅可以根据需要发布信息，也可以根据自己的喜好回复或转发信息，网络大数据直接反映了社会状态。



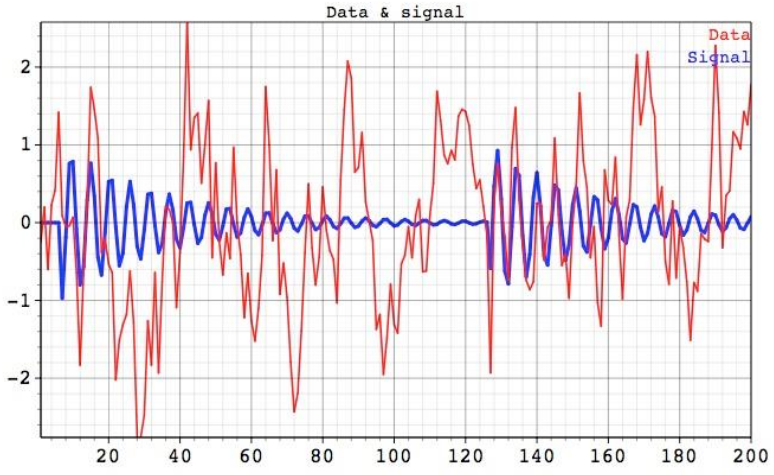
互联网数据

突发性：有些信息在传播过程中会在短时间内引起大量新的网络数据的产生，并使相关的网络用户形成网络群体，体现出网络大数据以及网络群体的突发特性。

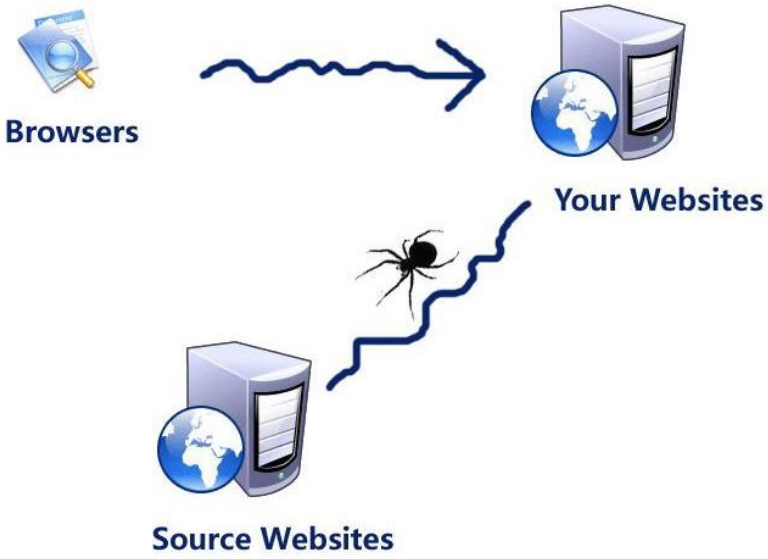


互联网数据

高噪音：网络大数据来自于众多不同的网络用户，具有很高的噪声和不确定性。



网络爬虫



网络爬虫是一种自动搜集互联网信息的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。

通过实现爬虫程序，可以搜集某一站点的URLs（网页地址），并将搜集到的URLs存入数据库。不仅能够为搜索引擎采集网络信息，而且可以作为定向信息采集器，采集某些网站下的特定信息。

网络爬虫

在网络爬虫的系统框架中，主过程由控制器，解析器，资源库三部分组成。

控制器： 控制器的主要工作是负责给多线程中的各个爬虫线程分配工作任务。

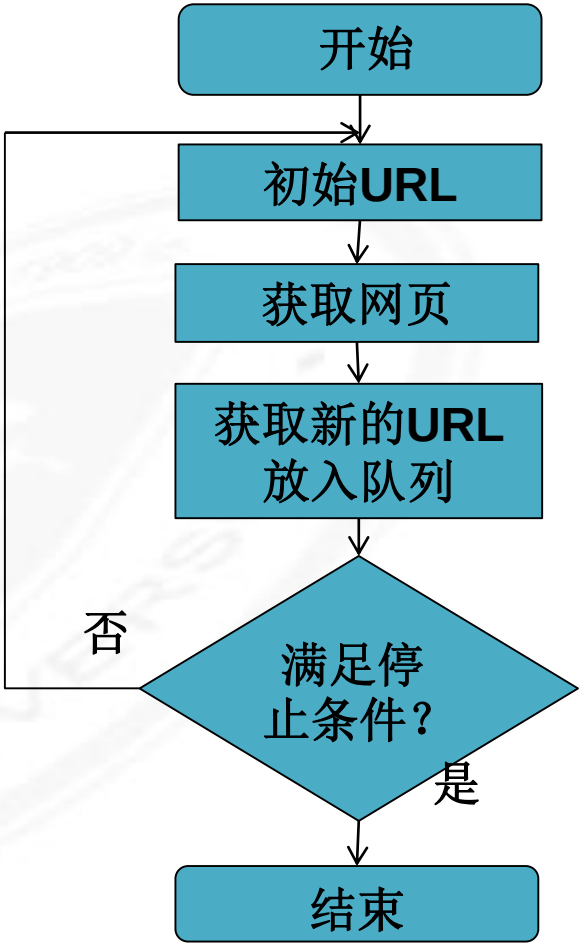
解析器： 解析器的主要工作是下载网页，进行页面的处理，主要是将一些JS脚本标签、CSS代码内容、空格字符、HTML标签等内容处理掉，爬虫的基本工作是由解析器完成。

资源库： 资源库是用来存放下载到的网页资源，一般都采用大型的数据库存储，如Oracle数据库，并对其建立索引。

网络爬虫

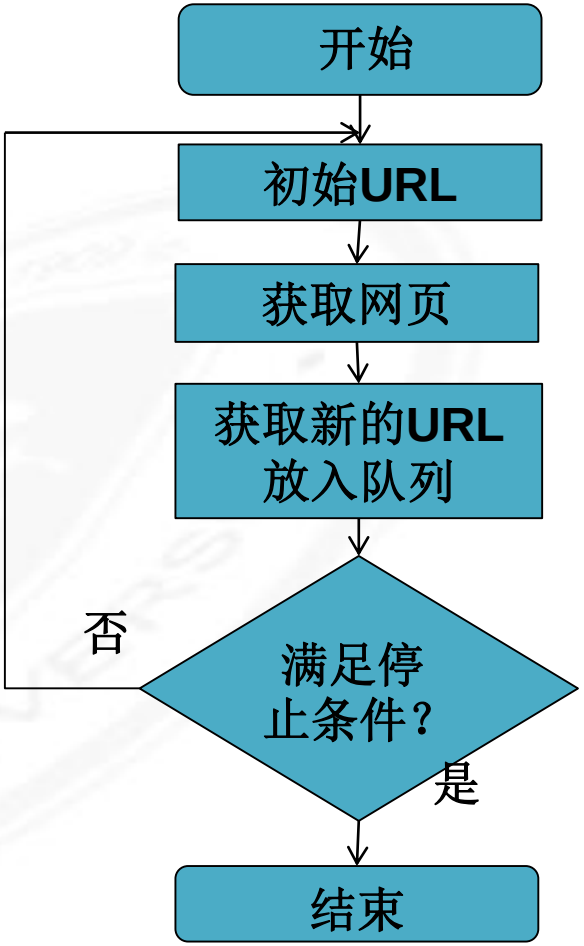
- **页面采集模块：**该模块是爬虫和因特网的接口，主要作用是通过各种 web 协议(一般以 HTTP．FTP 为主)来完成对网页数据的采集，保存后将采集到的页面交由后续模块作进一步处理。

其过程类似于用户使用浏览器打开网页，保存的网页供其它后续模块处理，例如，页面分析、链接抽取。



网络爬虫

- 页面分析模块：该模块的主要功能是将页面采集模块采集下来的页面进行分析，提取其中满足用户要求的超链接，加入到超链接队列中。

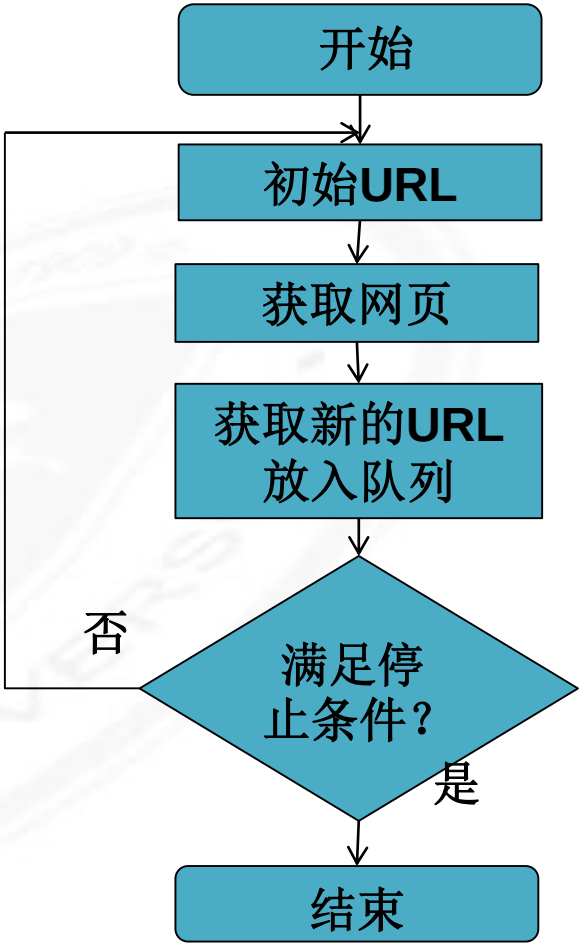


页面链接中给出的 URL 一般是多种格式的，可能是完整的包括协议、站点和路径的，也可能是省略了部分内容的，或者是一个相对路径。所以为处理方便，一般进行规范化处理，先将其转化成统一的格式。

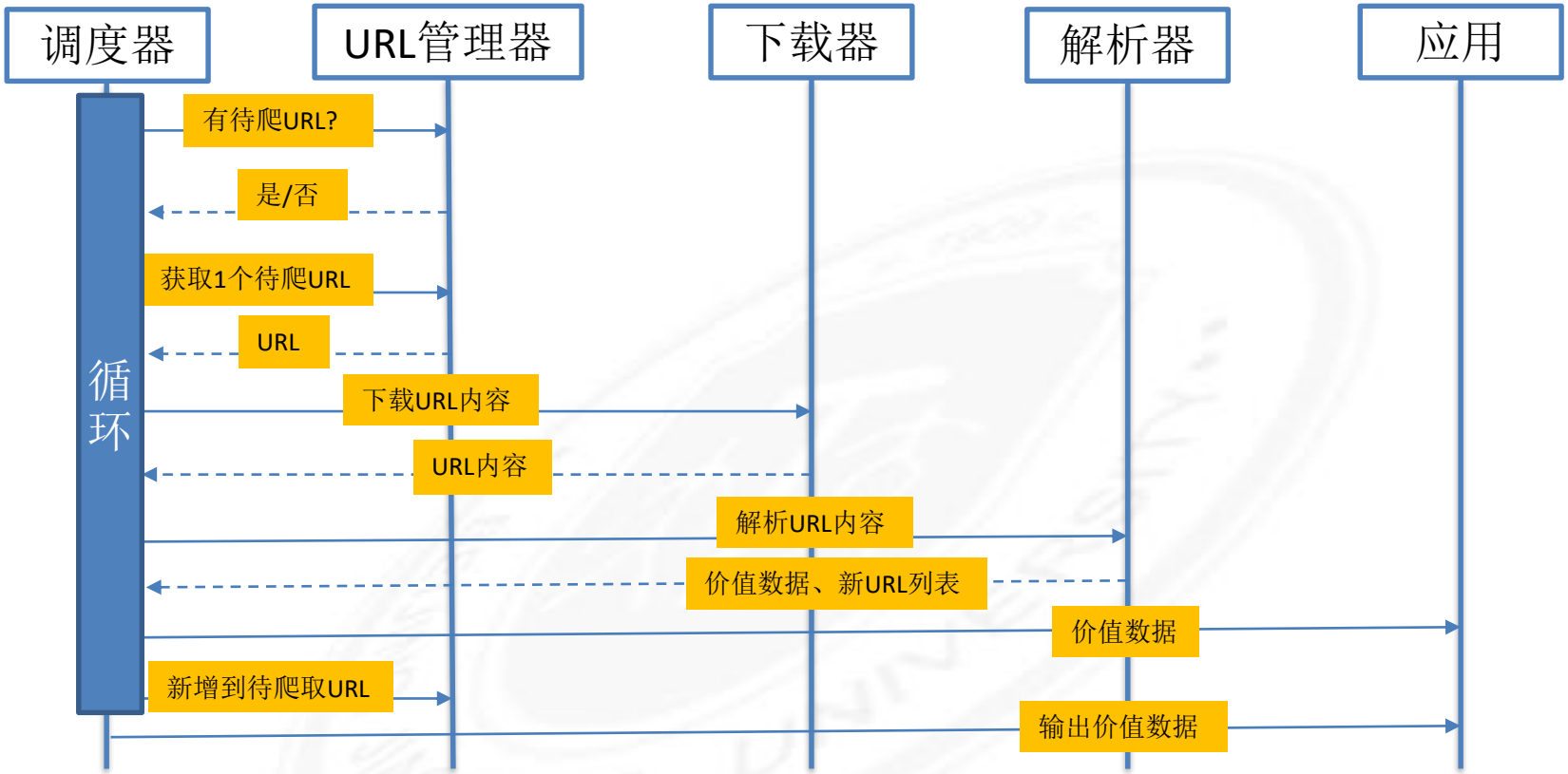
网络爬虫

- 链接过滤模块：该模块主要是用于对重复链接和循环链接的过滤。例如，相对路径需要补全 URL ，然后加入到待采集 URL 队列中。

此时，一般会过滤掉队列中已经包含的 URL，以及循环链接的URL。



网络爬虫


调度器
URL管理器
下载器
解析器
应用
循环
有待爬URL?
是/否
获取1个待爬URL
URL
下载URL内容
URL内容
解析URL内容
价值数据、新URL列表
价值数据
新增到待爬取URL
输出价值数据

网络爬虫

网络爬虫 URL 抓取策略有：

- 深度优先策略
- 广度优先策略
- Partial PageRank策略
- OPIC策略策略

深度优先策略

- 从URL池中选择某URL，然后按深度优先遍历以该URL为根节点的所有URL网页内容，然后取出URL池中下一个URL，继续上述策略循环至URL池遍历完。
- 优点是设计简易，抓取深度大
- 缺点是容易导致无限抓取，使得爬取过程无法收敛

广度优先策略

- 按照广度优先搜索思想，逐层抓取URL池中的每一个URL内容并将每一层的扇出URL纳入URL池中，按照广度优先策略循环遍历。
- 优点是抓取宽度广，抓取过程容易控制，有效减轻服务器的负载
- 缺点是容易造成URL大量聚集而导致URL池溢出

Partial PageRank策略

- Partial PageRank策略借鉴了PageRank算法的思想：对于已经下载的网页，连同待抓取URL队列中的URL，形成网页集合，计算每个页面的PageRank值，计算完之后，将待抓取URL队列中的URL按照PageRank值的大小排列，并按照该顺序抓取页面。
- 优点是抓取优先级可控
- 缺点是由于广告和作弊链接的存在，容易导致PageRank值不能完全刻画其重要程度，从而导致实际抓取数据无效

OPIC策略策略

- IPIC: Online Page Importance Computation
- 该算法实际上也是对页面进行一个重要性打分。在算法开始前，给所有页面一个相同的初始现金（cash）。当下载了某个页面P之后，将P的现金分摊给所有从P中分析出的链接，并且将P的现金清空。对于待抓取URL队列中的所有页面按照现金数进行排序。
- 优点是计算速度快于Partial PageRank策略，适合实时计算场景
- 缺点是受初始值影响大

网络爬虫工具

Python

Java

Larbin

C++

小作业

简答题:

如果给你一个网站，你要怎么爬取里面的有用内容？

（阐述主要思路和大致方案即可，不需要实现）

提交方式:

课程平台，word格式

提交时间:

下周六（5.14）