# 关键技术研究基础与现状

## 2.1 网络爬虫技术概述

### 2.1.1 HTTP协议简介

### 2.1.2网络爬虫抓取策略

2.1.3分布式在网络爬虫技术中的应用

2.2基于分布式的大数据处理技术

### 2.2.1大数据和分布式计算

### 2.2.2 MapReduce和分布式存储技术

### 2.2.3大数据处理平台Hadoop简介

## 2.3基于DPI数据的Web用户行为分析

### 2.3.1 Web用户行为概述

### 2.3.2 DPI简介

### 2.3.3网络访问流量分析工具WireShark介绍

### 2.3.4 Web用户行为分析与用户画像构建

## 2.4 用户兴趣推荐系统概述

## 2.4.1 用户兴趣推荐系统的意义

## 2.4.2 常见推荐模型

## 2.4.3 推荐系统的评价指标

# 第四章 基于Hadoop平台的电信DPI数据处理

## 4.1 电信DPI数据预处理

### 4.1.1电信DPI数据结构和数据规模

### 4.1.2电信DPI数据清洗与格式化

### 4.1.3用户会话识别

### 4.1.4电商用户数据过滤与分类

## 4.2电信DPI数据预处理的Hadoop编程实现

### 4.2.1 MapReduce和GFS简介

2003年到2004年间，Google发表了MapReduce、GFS（Google File System）和BigTable三篇技术论文，提出了一套全新的分布式计算理论。

MapReduce是分布式计算框架，GFS（Google File System）是分布式文件系统，BigTable是基于Google File System的数据存储系统，这三大组件组成了Google的分布式计算模型。Google的分布式计算模型相比于传统的分布式计算模型有三大优势：首先，它简化了传统的分布式计算理论，降低了技术实现的难度，可以进行实际的应用。其次，它可以应用在廉价的计算设备上，只需增加计算设备的数量就可以提升整体的计算能力，应用成本十分低廉。最后，它被Google应用在Google的计算中心，取得了很好的效果，有了实际应用的证明。

MapReduce处理过程原理如下：



#### MapReduce Library首先将输入文件切割成多个小片的文件pieces，然后MapReduce Library将启动复制操作，将用户程序复制到各个a cluster of machines上。

#### 在这些a cluster of machines中，其中一个比较特殊称之为master，其他的machine被称之为worker，master选择空闲的worker并将任务（map任务或者是reduce任务）分配给这些空闲的worker任务。

#### 一个worker如果被master分配了map任务的话，该worker首先读取该key/value对，然后执行用户定义的map函数，这些处理完成的key/value对被缓存到内存中。

#### 然后，将这些key/value对写入本地磁盘，然后worker通知master。

#### 如果master接收到了worker在第4步的通知之后，master将这个信息传递给reduce worker，该reduce worker通过远程系统调用的形式读取该worker磁盘上存储的处理完的数据。

#### Reduce worker遍历已排序的数据，然后将数据传递到用户定义的Reduce函数。

#### 当所有的map和reduce完成之后，然后master唤醒用户程序。

通过MapReduce运行流程可以看出，用户程序仅仅需要编写Map函数和Reduce函数即可，MapReduce库首先通过调用用户自定义的Map函数，将输入文件分割，如果数据处理完成，将调用Reduce函数将结果合并起来。这样大大简化了用户应用程序编写的复杂度。

。

### 4.2.2 Hadoop分布式计算框架原理

Hadoop采用MapReduce分布式计算框架，并根据GFS开发了HDFS分布式文件系统，根据BigTable开发了HBase数据存储系统。Hadoop的开源特性使其成为分布式计算系统的事实上的国际标准。Yahoo，Facebook，Amazon以及国内的百度，阿里巴巴等众多互联网公司都以Hadoop为基础搭建自己的分布式计算系统。

MapReduce是Google提出的一个软件架构，用于大规模数据集（大于1TB）的并行运算。概念“Map（映射）”和“Reduce（归纳）”，及他们的主要思想，都是从函数式编程语言以及矢量编程语言特性借鉴来的，暨通过用户定义的map函数将输入分割成key/value对，然后处理该数据，最终通过Reduce函数将处理完成的记过合并。