《数据采集与预处理》

教材官网： http://dblab.xmu.edu.cn/post/data-collection/

温馨提示：编辑幻灯片母版，可以修改每页PPT的厦大校徽和底部文字

第4章 分布式消息系统Kafka

（PPT版本号：2022年1月版本）

林子雨 副教授

厦门大学计算机科学与技术系

E-mail: ziyulin@xmu.edu.cn

主页：**http://dblab.xmu.edu.cn/linziyu**

提纲

4.1 Kafka简介
4.2 Kafka在大数据生态系统中的作用
4.3 Kafka与Flume的区别与联系
4.4 Kafka相关概念
4.5 Kafka的安装和使用
4.6 使用Python操作Kafka
4.7 Kafka与MySQL的组合使用

本PPT是以下教材的配套讲义
林子雨编著《数据采集与预处理》
人民邮电出版社

教材官网：
http://dblab.xmu.edu.cn/post/data-collection

4.1 Kafka简介

4.1.1 Kafka的特性
4.1.2 Kafka的应用场景
4.1.3 Kafka的消息传递模式

本PPT是以下教材的配套讲义
林子雨编著《数据采集与预处理》
人民邮电出版社

教材官网：
http://dblab.xmu.edu.cn/post/data-collection

4.1.1 Kafka的特性

Kafka具有以下良好的特性：

•高吞吐量、低延迟：Kafka每秒可以处理几十万条消息，它的延迟最低只有几毫秒；

•可扩展性：Kafka集群具有良好的可扩展性；

•持久性、可靠性：消息被持久化到本地磁盘，并且支持数据备份，防止数据丢失；

•容错性：允许集群中节点失败（若副本数量为n，则允许n-1个节点失败）；

•高并发：支持数千个客户端同时读写。使用消息队列能够使关键组件顶住突发的访问压力，而不会因为突发的超负荷的请求而完全崩溃；

•顺序保证：在大多使用场景下，数据处理的顺序都很重要。大部分消息队列本来就是排序的，并且能保证数据会按照特定的顺序来处理。Kafka保证一个分区内的消息的有序性；

•异步通信：很多时候，用户不想也不需要立即处理消息。消息队列提供了异步处理机制，允许用户把一个消息放入队列，但并不立即处理它。想向队列中放入多少消息就放多少，然后在需要的时候再去处理它们。

4.1.2 Kafka的应用场景

Kafka的主要应用场景包括：

•日志收集：一个公司可以用Kafka收集各种日志，这些日志被Kafka收集以后，可以通过Kafka的统一接口服务开放给各种消费者，例如Hadoop、HBase、Solr等；

•消息系统：可以对生产者和消费者实现解耦，并可以缓存消息；

•用户活动跟踪：Kafka经常被用来记录Web用户或者APP用户的各种活动，如浏览网页、搜索、点击等活动，这些活动信息被各个服务器发布到Kafka的主题（Topic）中，然后订阅者通过订阅这些主题来做实时的监控分析，或者装载到Hadoop、数据仓库中做离线分析和挖掘；

•运营指标：Kafka也经常用来记录运营监控数据，包括收集各种分布式应用的数据，生产环节各种操作的集中反馈，比如报警和报告；

•流式处理：Kafka实时采集的数据可以传递给流处理框架（比如Spark Streaming和Storm）进行实时处理。

4.1.3 Kafka的消息传递模式

一个消息系统负责将数据从一个应用传递到另外一个应用，应用只需关注于数据，无需关注数据在两个或多个应用间是如何传递的。分布式消息传递基于可靠的消息队列，在客户端应用和消息系统之间异步传递消息。对于消息系统而言，一般有两种主要的消息传递模式：点对点传递模式和发布订阅模式。大部分的消息系统选用发布订阅模式。Kafka就是一种发布订阅模式。

1. 点对点消息传递模式

在点对点消息系统中（如图4-1所示），消息持久化到一个队列中。此时，将有一个或多个消费者消费队列中的数据。但是一条消息只能被消费一次。当一个消费者消费了队列中的某条数据之后，该条数据则从消息队列中删除。该模式即使有多个消费者同时消费数据，也能保证数据处理的顺序。

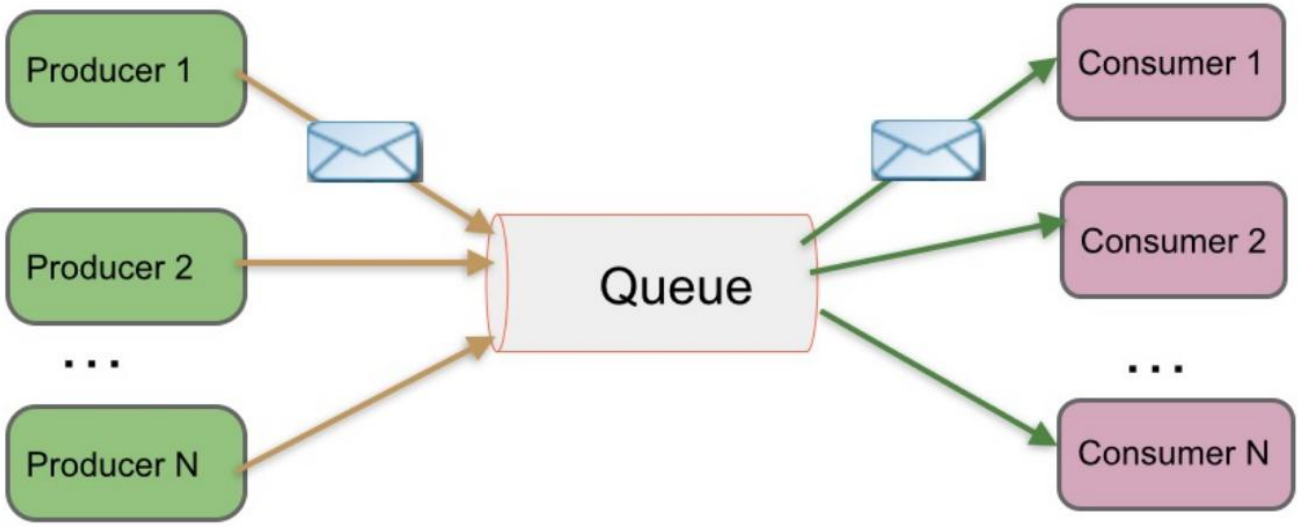


图4-1 点对点消息系统的架构

2. 发布订阅消息传递模式

在发布订阅消息系统中（如图4-2所示），消息被持久化到一个主题（topic）中。与点对点消息系统不同的是，消费者可以订阅一个或多个主题，消费者可以消费该主题中所有的数据，同一条数据可以被多个消费者消费，数据被消费后不会立马删除。在发布订阅消息系统中，消息的生产者称为“发布者”，消费者称为“订阅者”。

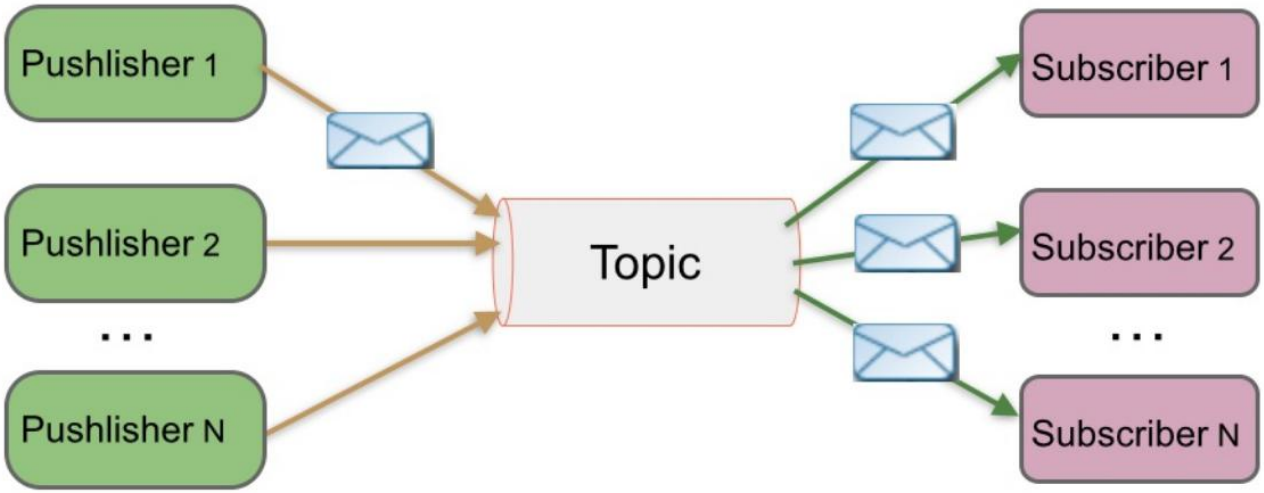


图4-2 发布订阅消息传递模式的架构

4.2 Kafka在大数据生态系统中的作用

如图4-3所示，在公司的大数据生态系统中，可以把Kafka作为数据交换枢纽，不同类型的分布式系统（关系数据库、NoSQL数据库、流处理系统、批处理系统等），可以统一接入到Kafka，实现和Hadoop各个组件之间的不同类型数据的实时高效交换，较好地满足各种企业应用需求。同时，借助于Kafka作为交换枢纽，也可以很好解决不同系统之间的数据生产/消费速率不同的问题。比如在线上实时数据需要写入HDFS的场景中，线上数据不仅生成速率快，而且具有突发性，如果直接把线上数据写入HDFS，可能会导致高峰时间HDFS写入失败，在这种情况下，就可以先把线上数据写入Kafka，然后借助于Kafka导入到HDFS。

4.2 Kafka在大数据生态系统中的作用

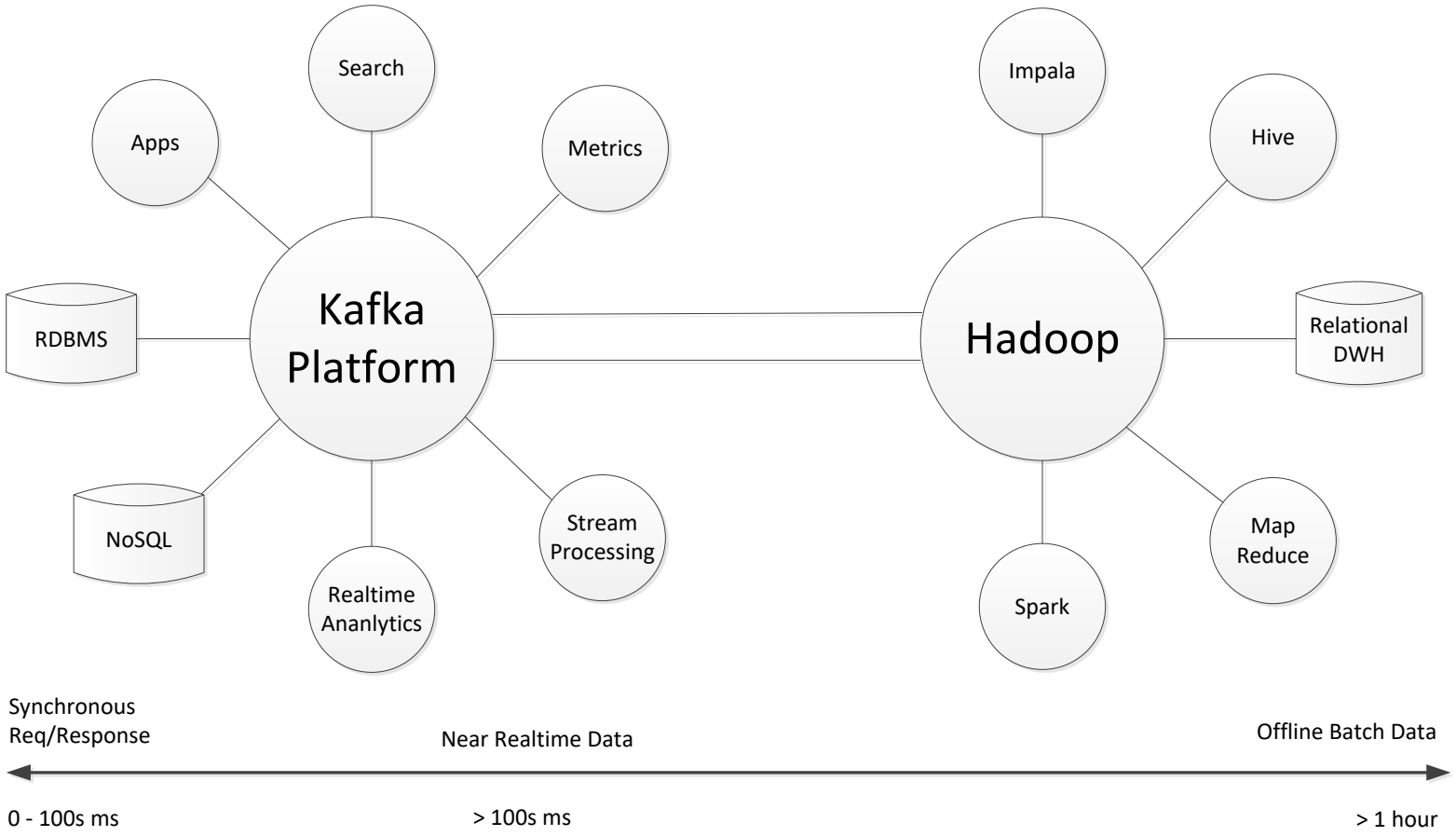


图4-3 Kafka作为数据交换枢纽

4.3 Kafka与Flume的区别与联系

Kafka与Flume的很多功能确实是重叠的，二者的联系与区别如下：

（1）Kafka是一个通用型系统，可以有许多的生产者和消费者分享多个主题。相反地，Flume被设计成特定用途的工作，特定地向 HDFS 和 HBase 发送数据。Flume为了更好地为 HDFS 服务而做了特定的优化，并且与 Hadoop 的安全体系整合在了一起。因此，如果数据需要被多个应用程序消费的话，推荐使用 Kafka，如果数据只是面向 Hadoop 的，可以使用 Flume。

（2）Flume拥有许多配置的数据源 (source) 和数据槽(sink)，而Kafka拥有的是非常小的生产者和消费者环境体系。如果数据来源已经确定，不需要额外的编码，那么可以使用Flume 提供的数据源和数据槽，反之，如果需要准备自己的生产者和消费者，那么就需要使用Kafka。

（3）Flume可以在拦截器里面实时处理数据，这个特性对于过滤数据非常有用。Kafka需要一个外部系统帮助处理数据。

（4）无论是Kafka或是Flume，两个系统都可以保证不丢失数据。

（5）Flume和Kafka可以一起工作。Kafka是分布式消息中间件，自带存储，更合适做日志缓存，Flume数据采集部分做得很好，可以使用Flume采集日志，然后，把采集到的日志发送到Kafka中，再由Kafka把数据传送给Hadoop、Spark等消费者。

4.4 Kafka相关概念

Kafka是一种高吞吐量的分布式发布订阅消息系统，为了更好地理解和使用Kafka，这里介绍一下Kafka的相关概念：

•Broker：Kafka集群包含一个或多个服务器，这些服务器被称为“Broker”。

•Topic：每条发布到Kafka集群的消息都有一个类别，这个类别被称为“Topic（主题）”。物理上不同Topic的消息分开存储，逻辑上一个Topic的消息虽然保存于一个或多个Broker上，但用户只需指定消息的Topic，即可生产或消费数据，而不必关心数据存于何处。

•Partition：是物理上的概念，每个Topic包含一个或多个Partition。

•Producer：负责发布消息到Kafka Broker。

•Consumer：消息消费者，向Kafka Broker读取消息的客户端。

•Consumer Group：每个Consumer属于一个特定的Consumer Group，可为每个Consumer指定Group Name，若不指定Group Name，则属于默认的Group。同一个Topic的一条消息只能被同一个Consumer Group内的一个Consumer消费，但多个Consumer Group可同时消费这一消息。

4.4 Kafka相关概念

图4-4给出了Kafka的总体架构。一个典型的Kafka集群中包含若干Producer、若干Broker、若干Consumer以及一个Zookeeper集群。Kafka通过Zookeeper管理集群配置。Producer使用push模式将消息发布到Broker，Consumer使用pull模式从Broker订阅并消费消息。

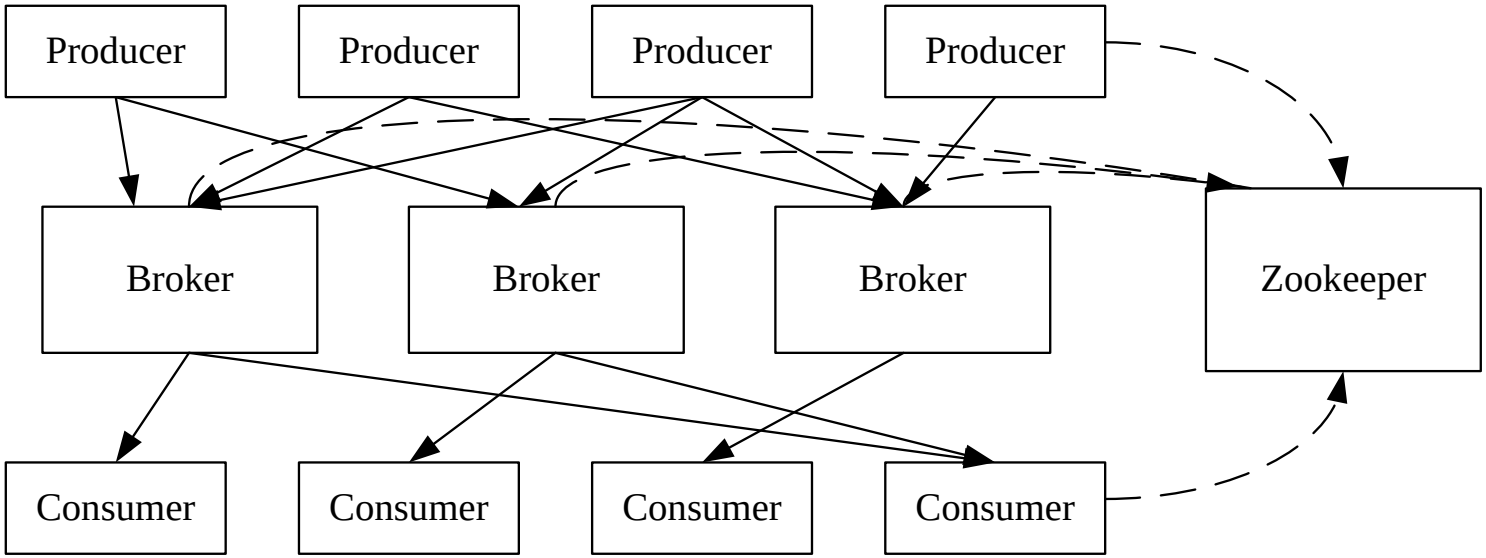


图4-4 Kafka总体架构

4.4 Kafka相关概念

图4-5描述了Kafka中的Topic与其他组件的关系。Producer在发布消息时，会发布到特定的Topic，Consumer是从特定的Topic获取消息。每个Topic包含一个或多个Partition。

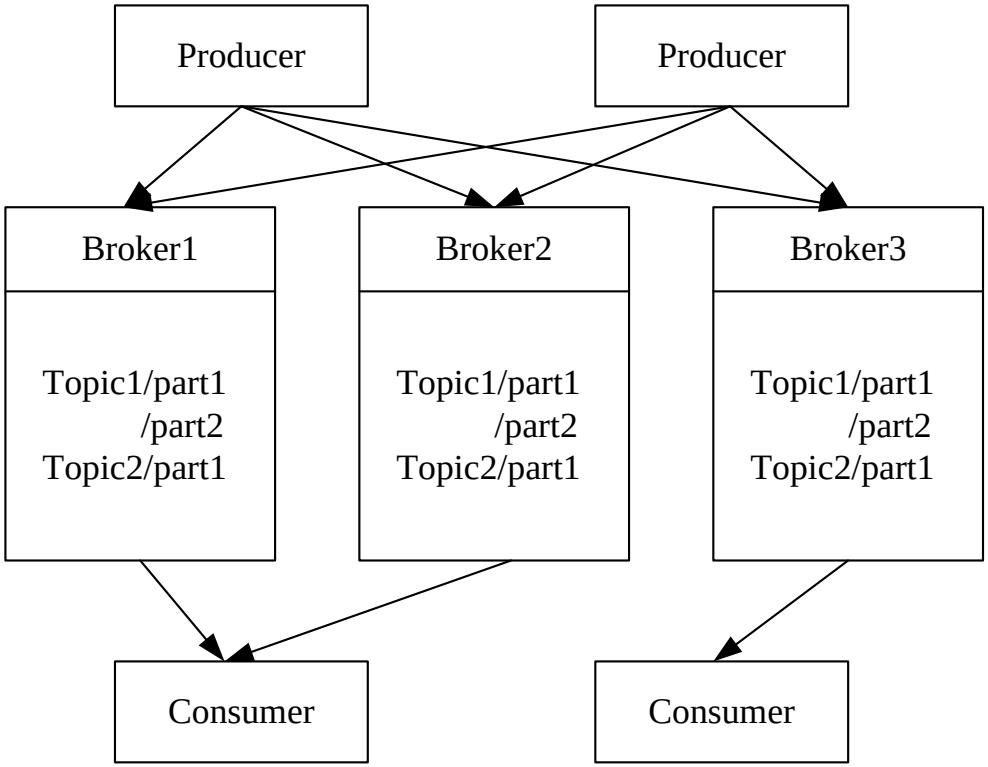


图4-5 Kafka中的Topic

4.5 Kafka的安装和使用

4.5.1 安装Kafka
4.5.2 使用Kafka

4.5.1 安装Kafka

Kafka的运行需要Java环境的支持，因此，需要在Windows系统中安装JDK。请参照第2章内容完成JDK的安装。
访问Kafka官网（http://kafka.apache.org/downloads），下载Kafka2.4.0版本的安装文件kafka_2.12-2.4.0.tgz，解压缩到“C:\”下。
因为Kafka的运行需要依赖于Zookeeper，因此，需要下载并安装Zookeeper。当然，Kafka也内置了Zookeeper服务，因此，也可以不用额外安装Zookeeper，而是直接使用内置的Zookeeper服务。为了简单起见，这里直接使用Kafka内置的Zookeeper服务。

4.5.2 使用Kafka

在Windows系统中打开第1个cmd窗口，启动Zookeeper服务：

```
> cd c:\kafka_2.12-2.4.0
> .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.Properties
```

注意，执行上面命令以后，cmd窗口会返回一堆信息，然后就停住不动了，没有回到命令提示符状态，这时，不要误以为死机了，而是Zookeeper服务器已经启动，正在处于服务状态。所以，不要关闭这个cmd窗口，一旦关闭，Zookeeper服务就停止了。

4.5.2 使用Kafka

打开第2个cmd窗口，然后输入下面命令启动Kafka服务：
> cd c:\kafka_2.12-2.4.0
> .\bin\windows\kafka-server-start.bat .\config\server.properties
执行上面命令以后，如果启动失败，并且出现提示信息“此时不应有\QuickTime\QTSystem\QTJava.zip”，则需要把CLASSPATH环境变量的相关信息删除，具体方法是：右键点击“我的电脑”->“高级系统设置”->“环境变量”，然后，找到CLASSPATH环境变量，把类似如下的信息删除：
C:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip
然后重新启动计算机，让配置修改生效。重新启动计算机以后，再次按照上面方法启动Zookeeper和Kafka。

执行上面命令以后，如果启动成功，cmd窗口会返回一堆信息，然后就会停住不动，没有回到命令提示符状态，这时，同样不要误以为死机了，而是Kafka服务器已经启动，正在处于服务状态。所以，不要关闭这个cmd窗口，一旦关闭，Kafka服务就停止了。

4.5.2 使用Kafka

为了测试Kafka，这里创建一个主题（Topic），名称为“topic_test”，包含一个分区，只有一个副本，在第3个cmd窗口中执行如下命令：

```
> cd c:\kafka_2.12-2.4.0
> .\bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic topic_test
```

可以继续执行如下命令，查看topic_test是否创建成功：

```
> .\bin\windows\kafka-topics.bat  --list  --zookeeper  localhost:2181
```

如果创建成功，就可以在执行结果中看到topic_test。

4.5.2 使用Kafka

继续在第3个cmd窗口中执行如下命令创建一个生产者来产生消息：

> .\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic topic_test

该命令执行以后，屏幕上的光标会一直在闪烁，这时，就可以用键盘输入一些内容，比如输入：

I love Kafka

Kafka is good

新建第4个cmd窗口，执行如下命令来消费消息：

> cd c:\kafka_2.12-2.4.0

> .\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic topic_test --from-beginning

该命令执行以后，就会在屏幕上看到刚才输入的语句“I love Kafka”和“Kafka is good”。

4.6 使用Python操作Kafka

使用Python操作Kafka之前，需要安装第三方模块python-kafka，命令如下：

```
> pip install kafka-python
```

安装结束以后，可以使用如下命令查看已经安装的kafka-python的版本信息：

```
> pip list
```

这个命令会显示已经安装的Pyhon第三方模块，并且给出每个模块的版本信息。

4.6 使用Python操作Kafka

编写一个生产者程序producer_test.py用来生成消息：

```
from kafka import KafkaProducer

producer = KafkaProducer(bootstrap_servers='localhost:9092')  # 连接Kafka

msg = "Hello World".encode('utf-8')  # 发送内容,必须是bytes类型
producer.send('test', msg)  # 发送的topic为test
producer.close()
```

4.6 使用Python操作Kafka

编写一个消费者程序consumer_test.py用来消费消息：

```
from kafka import KafkaConsumer

consumer = KafkaConsumer('test',
bootstrap_servers=['localhost:9092'],group_id=None,auto_offset_res
et='smallest')
for msg in consumer:
    recv = "%s:%d:%d: key=%s value=%s" % (msg.topic,
msg.partition, msg.offset, msg.key, msg.value)
    print(recv)
```

启动Zookeeper服务和Kafka服务，然后，先执行producer_test.py，再执行consumer_test.py，就可以看到屏幕上打印出“Hello World”。

4.6 使用Python操作Kafka

下面再给出一个稍微复杂一点的实例。假设有一个文件score.csv，其内容如下：

```
"Name","Score"
"Zhang San",99.0
"Li Si",45.5
"Wang Hong",82.5
"Liu Qian",76.0
"Ma Li",62.5
"Shen Teng",78.0
"Pu Wen",86.5
```

4.6 使用Python操作Kafka

要求完成的任务是，Kafka生产者读取文件中的所有内容，然后，以JSON字符串的形式发送给Kafka消费者，消费者获得消息以后转换成表格形式打印到屏幕上，如下所示：

```
      Name  Score
0  Zhang San   99.0
1      Li Si   45.5
2  Wang Hong   82.5
3   Liu Qian   76.0
4      Ma Li   62.5
5  Shen Teng   78.0
6     Pu Wen   86.5
```

4.6 使用Python操作Kafka

为了完成上述任务，可以编写代码文件kafka_demo.py（要求和文件score.csv在同一个目录下），其内容如下：

```
# kafka_demo.py
import sys
import json
import pandas as pd
import os
from kafka import KafkaProducer
from kafka import KafkaConsumer
from kafka.errors import KafkaError

KAFKA_HOST = "localhost" #服务器地址
KAFKA_PORT = 9092    #端口号
KAFKA_TOPIC = "topic0"  #topic

data=pd.read_csv(os.getcwd()+'\\score.csv')
key_value=data.to_json()
```

4.6 使用Python操作Kafka

```
class Kafka_producer():
    def __init__(self, kafkahost, kafkaport, kafkatopic, key):
      self.kafkaHost = kafkahost
      self.kafkaPort = kafkaport
      self.kafkatopic = kafkatopic
      self.key = key
      self.producer =
KafkaProducer(bootstrap_servers='{kafka_host}:{kafka_port}'.format(
          kafka_host=self.kafkaHost,
          kafka_port=self.kafkaPort)
       )
    def sendjsondata(self, params):
        try:
           parmas_message = params
           producer = self.producer
           producer.send(self.kafkatopic, key=self.key,
value=parmas_message.encode('utf-8'))
           producer.flush()
        except KafkaError as e:
            print(e)
```

4.6 使用Python操作Kafka

```
class Kafka_consumer():
    def __init__(self, kafkahost, kafkaport, kafkatopic, groupid,key):
        self.kafkaHost = kafkahost
        self.kafkaPort = kafkaport
        self.kafkatopic = kafkatopic
        self.groupid = groupid
        self.key = key
        self.consumer = KafkaConsumer(self.kafkatopic, group_id=self.groupid,
            bootstrap_servers='{kafka_host}:{kafka_port}'.format(
            kafka_host=self.kafkaHost,
            kafka_port=self.kafkaPort)
         )
    def consume_data(self):
        try:
            for message in self.consumer:
                yield message
        except KeyboardInterrupt as e:
                print(e)
```

4.6 使用Python操作Kafka

```
def sortedDictValues(adict):
    items = adict.items()
    items=sorted(items,reverse=False)
    return [value for key, value in items]
```

4.6 使用Python操作Kafka

```
def main(xtype, group, key):
    if xtype == "p":
        # 生产模块
        producer = Kafka_producer(KAFKA_HOST, KAFKA_PORT, KAFKA_TOPIC, key)
        print("===========> producer:", producer)
        params =key_value
        producer.sendjsondata(params)
    if xtype == 'c':
        # 消费模块
        consumer = Kafka_consumer(KAFKA_HOST, KAFKA_PORT, KAFKA_TOPIC, group,key)
        print("===========> consumer:", consumer)
        message = consumer.consume_data()
        for msg in message:
            msg=msg.value.decode('utf-8')
            python_data=json.loads(msg) ##字符串转换成字典
            key_list=list(python_data)
            test_data=pd.DataFrame()
            for index in key_list:
                if index=='Name':
                    a1=python_data[index]
                    data1 = sortedDictValues(a1)
                    test_data[index]=data1
                else:
                    a2 = python_data[index]
                    data2 = sortedDictValues(a2)
                    test_data[index] = data2
            print(test_data)
```

4.6 使用Python操作Kafka

```
if __name__ == '__main__':
        main(xtype='p',group='py_test',key=None)
        main(xtype='c',group='py_test',key=None)
```

4.7 Kafka与MySQL的组合使用

这里通过一个实例来演示Kafka与MySQL的组合使用。需要完成的任务是，把JSON格式数据放入Kafka发送出去，然后，再从Kafka中获取到JSON格式数据，对其进行解析并写入到MySQL数据库。请参照第2章的内容完成MySQL数据库的安装，并学习其使用方法。

4.7 Kafka与MySQL的组合使用

编写一个生产者程序producer_json.py：

```
# producer_json.py
from kafka import KafkaProducer
import json

producer =
KafkaProducer(bootstrap_servers='localhost:9092',value_serializer=lamb
da v:json.dumps(v).encode('utf-8'))  # 连接kafka

data={
  "sno":"95001",
  "name":"John",
  "sex":"M",
  "age":23
 }

producer.send('json_topic', data)  # 发送的topic为json_topic
producer.close()
```

4.7 Kafka与MySQL的组合使用

编写一个消费者程序consumer_json.py：

```
# consumer_json.py
from kafka import KafkaConsumer
import json
import pymysql.cursors

consumer = KafkaConsumer('json_topic',
bootstrap_servers=['localhost:9092'],group_id=None,auto_offset_reset='earliest')
for msg in consumer:
    msg1=str(msg.value, encoding = "utf-8")    #字节数组转成字符串
    dict = json.loads(msg1)    #字符串转换成字典
    # 连接数据库
    connect = pymysql.Connect(
    host='localhost',
    port=3306,
    user='root',  # 数据库用户名
    passwd='123456',  # 密码
    db='school',
    charset='utf8'
    )
```

4.7 Kafka与MySQL的组合使用

```
 # 获取游标
    cursor = connect.cursor()

    # 插入数据
    sql = "INSERT INTO student(sno,sname,ssex,sage) VALUES ('%s',
'%s', '%s', %d)"
    data = (dict['sno'],dict['name'],dict['sex'],dict['age'])
    cursor.execute(sql % data)
    connect.commit()
    print('成功插入数据')

    # 关闭数据库连接
    connect.close()
```

4.7 Kafka与MySQL的组合使用

在Windows系统中启动MySQL服务，然后，打开MySQL数据库的命令行界面，输入如下SQL语句创建数据库school：

```
mysql> CREATE DATABASE school;
```

创建好数据库school以后，可以使用如下SQL语句打开数据库：

```
mysql> USE school;
```

使用如下SQL语句创建一个表student：

```
mysql>CREATE TABLE student(
    -> sno char(5),
    -> sname char(10),
    -> ssex char(2),
    -> sage int);
```

4.7 Kafka与MySQL的组合使用

使用如下SQL语句查看已经创建的表：

```
mysql> SHOW TABLES;
```

在Windows系统中启动Zookeeper服务和Kafka服务，然后，先执行生产者程序producer_json.py，再执行消费者程序consumer_json.py，执行成功以后，使用如下命令查看MySQL数据库中新插入的记录：

```
mysql> SELECT * FROM student;
```

可以看到，一条记录已经被成功地插入到了MySQL数据库。

4.8 本章小结

Kafka是一个分布式、分区的、多副本的、多订阅者、基于Zookeeper协调的分布式日志系统，主要应用场景是日志收集系统和消息系统。LinkedIn于2010年把Kafka贡献给了Apache基金会并成为顶级开源项目。Kafka能够以时间复杂度为O(1)的方式提供消息持久化能力，即使对TB级以上数据也能保证常数时间的访问性能。Kafka还支持高吞吐率，即使在非常廉价的商用机器上也能做到单机支持每秒10万条消息的传输。本章内容介绍了Kafka的概念以及安装和使用方法。本章介绍的Kafka使用方法较为基础，要想了解更多高级的使用方法，读者可以参考相关书籍或网络资料。

附录A：主讲教师林子雨简介

主讲教师：林子雨

单位：厦门大学计算机科学与技术系
E-mail: ziyulin@xmu.edu.cn
个人网页：http://dblab.xmu.edu.cn/post/linziyu
数据库实验室网站：http://dblab.xmu.edu.cn

扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），全国高校知名大数据教师，现为厦门大学计算机科学系副教授，厦门大学信息学院实验教学中心主任，曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国计算机学会数据库专业委员会委员，中国计算机学会信息系统专业委员会委员。国内高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度、2017年度和2020年度厦门大学教学类奖教金获得者，荣获2019年福建省精品在线开放课程、2018年厦门大学高等教育成果特等奖、2018年福建省高等教育教学成果二等奖、2018年国家精品在线开放课程。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学青年基金项目(No.61303004)、1项福建省自然科学青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，主持的教改课题包括1项2016年福建省教改课题和1项2016年教育部产学协作育人项目，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。中国高校首个“数字教师”提出者和建设者，2009年至今，“数字教师”大平台累计向网络免费发布超过1000万字高价值的研究和教学资料，累计网络访问量超过1000万次。打造了中国高校大数据教学知名品牌，编著出版了中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》，并成为京东、当当网等网店畅销书籍；建设了国内高校首个大数据课程公共服务平台，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过400万次，累计访问量超过1500万次。

附录B：大数据学习路线图

大数据学习路线图访问地址：http://dblab.xmu.edu.cn/post/10164/

附录C：林子雨大数据系列教材

了解全部教材信息：http://dblab.xmu.edu.cn/post/bigdatabook/

附录D：《大数据导论（通识课版）》教材

开设全校公共选修课的优质教材

本课程旨在实现以下几个培养目标：

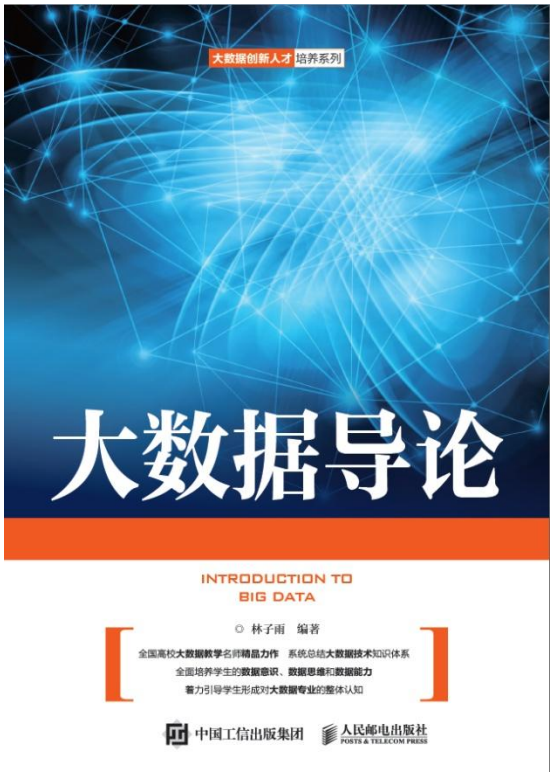
- 引导学生步入大数据时代，积极投身大数据的变革浪潮之中
- 了解大数据概念，培养大数据思维，养成数据安全意识
- 认识大数据伦理，努力使自己的行为符合大数据伦理规范要求
- 熟悉大数据应用，探寻大数据与自己专业的应用结合点
- 激发学生基于大数据的创新创业热情

高等教育出版社 ISBN:978-7-04-053577-8 定价：32元 版次：2020年2月第1版
教材官网：http://dblab.xmu.edu.cn/post/bigdataintroduction/

附录E：《大数据导论》教材

- 林子雨 编著《大数据导论》
- 人民邮电出版社，2020年9月第1版
- ISBN:978-7-115-54446-9 定价：49.80元

教材官网：http://dblab.xmu.edu.cn/post/bigdata-introduction/



开设大数据专业导论课的优质教材

扫一扫访问教材官网

附录F：《大数据技术原理与应用（第3版）》教材

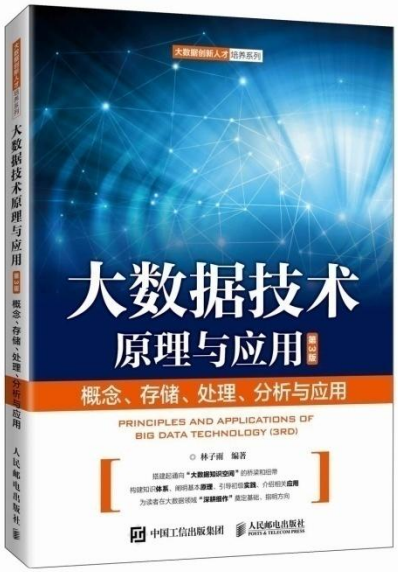
《大数据技术原理与应用——概念、存储、处理、分析与应用（第3版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社　ISBN:978-7-115-54405-6　定价：59.80元

全书共有17章，系统地论述了大数据的基本概念、大数据处理架构Hadoop、分布式文件系统HDFS、分布式数据 库HBase、NoSQL数据库、云数据库、分布式并行编程模型MapReduce、Spark、流计算、Flink、图计算、数据可视化以及大数据在互联网、生物医学和物流等各个领域的应用。在Hadoop、HDFS、HBase、MapReduce、Spark和Flink等重要章节，安排了入门级的实践操作，让读者更好地学习和掌握大数据关键技术。

本书可以作为高等院校计算机专业、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考、学习、培训之用。

欢迎访问《大数据技术原理与应用——概念、存储、处理、分析与应用》教材官方网站：
http://dblab.xmu.edu.cn/post/bigdata3

扫一扫访问教材官网



附录G：《大数据基础编程、实验和案例教程（第2版）》

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书

大数据教材

1+1黄金组合

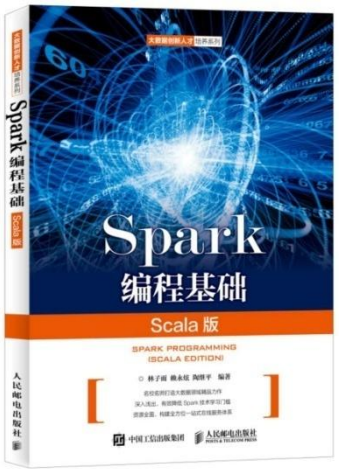
厦门大学林子雨编著

配套实验指导书

- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程

林子雨编著《大数据基础编程、实验和案例教程（第2版）》
清华大学出版社 ISBN:978-7-302-55977-1 定价：69元 2020年10月第2版

附录H：《Spark编程基础（Scala版）》



《Spark编程基础（Scala版）》

厦门大学 林子雨，赖永炫，陶继平 编著

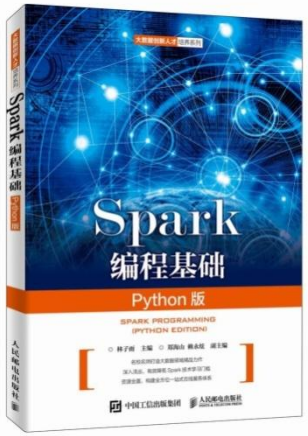
披荆斩棘，在大数据丛林中开辟学习捷径
填沟削坎，为快速学习Spark技术铺平道路
深入浅出，有效降低Spark技术学习门槛
资源全面，构建全方位一站式在线服务体系

人民邮电出版社出版发行，ISBN:978-7-115-48816-9

教材官网：http://dblab.xmu.edu.cn/post/spark/

本书以Scala作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Scala语言基础、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。

附录I：《Spark编程基础（Python版）》



《Spark编程基础（Python版）》

厦门大学 林子雨，郑海山，赖永炫 编著

披荆斩棘，在大数据丛林中开辟学习捷径
填沟削坎，为快速学习Spark技术铺平道路
深入浅出，有效降低Spark技术学习门槛
资源全面，构建全方位一站式在线服务体系

人民邮电出版社出版发行，ISBN:978-7-115-52439-3

教材官网： http://dblab.xmu.edu.cn/post/spark-python/

本书以Python作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、上机实验指南等。

附录J：高校大数据课程公共服务平台

高校大数据课程

公 共 服 务 平 台

http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/

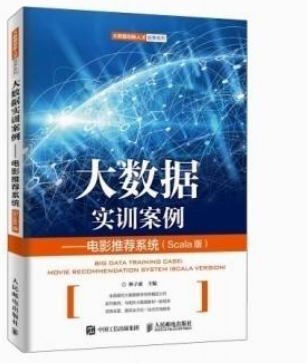
扫一扫访问平台主页

扫一扫观看3分钟FLASH动画宣传片

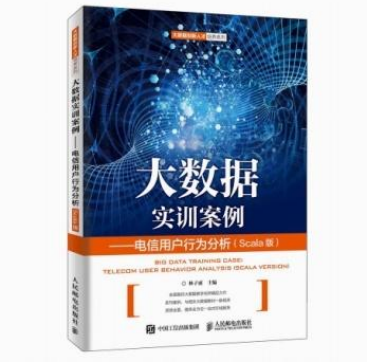
附录K：高校大数据实训课程系列案例教材

为了更好满足高校开设大数据实训课程的教材需求，厦门大学数据库实验室林子雨老师团队联合企业共同开发了《高校大数据实训课程系列案例》，目前已经完成开发的系列案例包括：

《电影推荐系统》（已经于2019年5月出版）
《电信用户行为分析》（已经于2019年5月出版）
《实时日志流处理分析》
《微博用户情感分析》
《互联网广告预测分析》
《网站日志处理分析》



系列案例教材将于2019年陆续出版发行，教材相关信息，敬请关注网页后续更新！
http://dblab.xmu.edu.cn/post/shixunkecheng/



扫一扫访问大数据实训课程系列案例教材主页


Thank You!