大数据专题技术型公开课

第2讲 分布式数据库HBase

林子雨 博士/助理教授

厦门大学计算机科学系

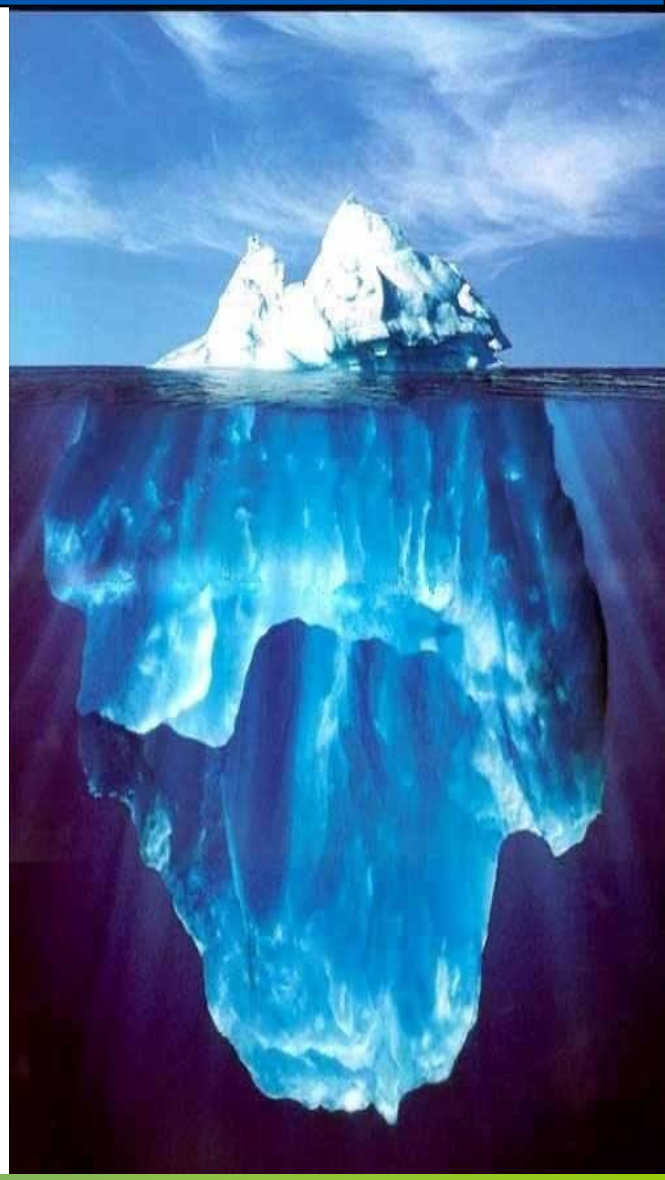
厦门大学云计算与大数据研究中心

E-mail: ziyulin@xmu.edu.cn

主页：http://www.cs.xmu.edu.cn/linziyu

提纲

- **4.1 概述**
- **4.2 HBase访问接口**
- **4.3 HBase数据模型**
- **4.4 HBase的实现原理**
- **4.5 HBase运行机制**
- **4.6 HBase编程实践**



本PPT是如下教材的配套讲义：
21世纪高等教育计算机规划教材
《大数据技术原理与应用
——概念、存储、处理、分析与应用》
（2015年6月第1版）
厦门大学 林子雨 编著，人民邮电出版社
ISBN:978-7-115-39287-9

欢迎访问《大数据技术原理与应用》教材官方网站：
http://dblab.xmu.edu.cn/post/bigdata

4.1 概述

- 4.1.1 从BigTable说起
- 4.1.2 HBase简介
- 4.1.3 HBase与传统关系数据库的对比分析

4.1.1从BigTable说起

- BigTable是一个分布式存储系统
- 利用谷歌提出的MapReduce分布式并行计算模型来处理海量数据
- 使用谷歌分布式文件系统GFS作为底层数据存储
- 采用Chubby提供协同服务管理
- 可以扩展到PB级别的数据和上千台机器，具备广泛应用性、可扩展性、高性能和高可用性等特点
- 谷歌的许多项目都存储在BigTable中，包括搜索、地图、财经、打印、社交网站Orkut、视频共享网站YouTube和博客网站Blogger等

4.1.2HBase简介

HBase是一个高可靠、高性能、面向列、可伸缩的分布式数据库，是谷歌BigTable的开源实现，主要用来存储非结构化和半结构化的松散数据。HBase的目标是处理非常庞大的表，可以通过水平扩展的方式，利用廉价计算机集群处理由超过10亿行数据和数百万列元素组成的数据表

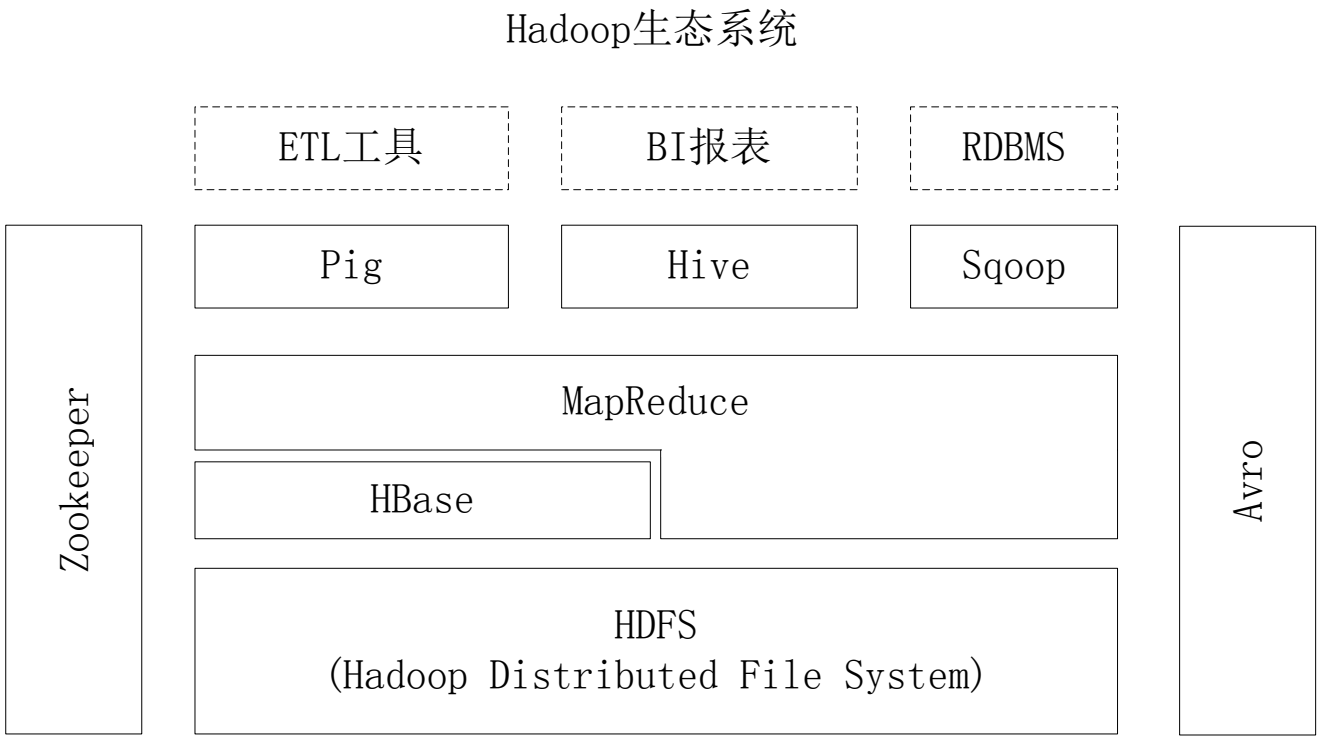


图4-1 Hadoop生态系统中HBase与其他部分的关系

4.1.2HBase简介

表4-1 HBase和BigTable的底层技术对应关系

	BigTable	**HBase**
文件存储系统	GFS	HDFS
海量数据处理	MapReduce	Hadoop MapReduce
协同服务管理	Chubby	Zookeeper

4.1.3HBase与传统关系数据库的对比分析

- HBase与传统的关系数据库的区别主要体现在以下几个方面：
- （1）数据类型：关系数据库采用关系模型，具有丰富的数据类型和存储方式，HBase则采用了更加简单的数据模型，它把数据存储为未经解释的字符串
- （2）数据操作：关系数据库中包含了丰富的操作，其中会涉及复杂的多表连接。HBase操作则不存在复杂的表与表之间的关系，只有简单的插入、查询、删除、清空等，因为HBase在设计上就避免了复杂的表和表之间的关系
- （3）存储模式：关系数据库是基于行模式存储的。HBase是基于列存储的，每个列族都由几个文件保存，不同列族的文件是分离的

4.1.3HBase与传统关系数据库的对比分析

- HBase与传统的关系数据库的区别主要体现在以下几个方面：
- （4）数据索引：关系数据库通常可以针对不同列构建复杂的多个索引，以提高数据访问性能。HBase只有一个索引——行键，通过巧妙的设计，HBase中的所有访问方法，或者通过行键访问，或者通过行键扫描，从而使得整个系统不会慢下来
- （5）数据维护：在关系数据库中，更新操作会用最新的当前值去替换记录中原来的旧值，旧值被覆盖后就不会存在。而在HBase中执行更新操作时，并不会删除数据旧的版本，而是生成一个新的版本，旧有的版本仍然保留
- （6）可伸缩性：关系数据库很难实现横向扩展，纵向扩展的空间也比较有限。相反，HBase和BigTable这些分布式数据库就是为了实现灵活的水平扩展而开发的，能够轻易地通过在集群中增加或者减少硬件数量来实现性能的伸缩

4.2 HBase访问接口

表4-2 HBase访问接口

类型	特点	场合
Native Java API	最常规和高效的访问方式	适合Hadoop MapReduce作业并行批处理HBase表数据
HBase Shell	HBase的命令行工具，最简单的接口	适合HBase管理使用
Thrift Gateway	利用Thrift序列化技术，支持C++、PHP、Python等多种语言	适合其他异构系统在线访问HBase表数据
REST Gateway	解除了语言限制	支持REST风格的Http API访问HBase
Pig	使用Pig Latin流式编程语言来处理HBase中的数据	适合做数据统计
Hive	简单	当需要以类似SQL语言方式来访问HBase的时候

4.3 HBase数据模型

- 4.3.1 数据模型概述
- 4.3.2 数据模型相关概念
- 4.3.3 数据坐标
- 4.3.4 概念视图
- 4.3.5 物理视图
- 4.3.6 面向列的存储

4.3.1数据模型概述

- HBase是一个稀疏、多维度、排序的映射表，这张表的索引是行键、列族、列限定符和时间戳
- 每个值是一个未经解释的字符串，没有数据类型
- 用户在表中存储数据，每一行都有一个可排序的行键和任意多的列
- 表在水平方向由一个或者多个列族组成，一个列族中可以包含任意多个列，同一个列族里面的数据存储在一起
- 列族支持动态扩展，可以很轻松地添加一个列族或列，无需预先定义列的数量以及类型，所有列均以字符串形式存储，用户需要自行进行数据类型转换
- HBase中执行更新操作时，并不会删除数据旧的版本，而是生成一个新的版本，旧有的版本仍然保留

4.3.2数据模型相关概念

- 表：HBase采用表来组织数据，表由行和列组成，列划分为若干个列族
- 行：每个HBase表都由若干行组成，每个行由行键（row key）来标识。访问表中的行只有三种方式：（1）通过单个行键访问；（2）通过一个行键的区间来访问；（3）全表扫描
- 列族：一个HBase表被分组成许多“列族”的集合，它是基本的访问控制单元
- 列限定符：列族里的数据通过列限定符（或列）来定位
- 单元格：在HBase表中，通过行、列族和列限定符确定一个“单元格”（cell），单元格中存储的数据没有数据类型，总被视为字节数组byte[]
- 时间戳：每个单元格都保存着同一份数据的多个版本，这些版本采用时间戳进行索引

4.3.2数据模型相关概念

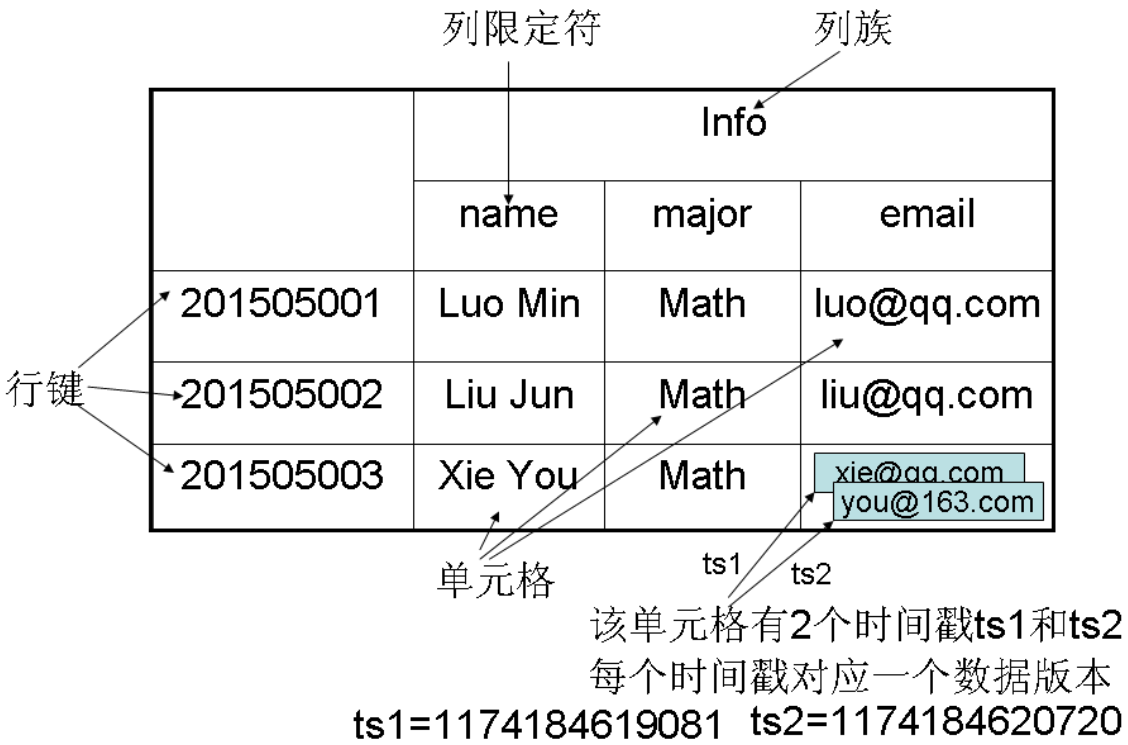


图4-2 HBase数据模型的一个实例

4.3.3数据坐标

- HBase中需要根据行键、列族、列限定符和时间戳来确定一个单元格，因此，可以视为一个“四维坐标”，即[行键, 列族, 列限定符, 时间戳]

键	值
[“201505003”, “Info”, “email”, 1174184619081]	“xie@qq.com”
[“201505003”, “Info”, “email”, 1174184620720]	“you@163.com”

4.3.4概念视图

表4-4 HBase数据的概念视图

行键	时间戳	列族contents	列族anchor
"com.cnn.www"	t5		anchor:cnnsi.com=”CNN”
	t4		anchor:my.look.ca="CNN.com"
	t3	contents:html="<html>..."	
	t2	contents:html="<html>..."	
	t1	contents:html="<html>..."	

4.3.5物理视图

表4-5 HBase数据的物理视图

列族contents

行键	时间戳	列族contents
"com.cnn.www"	t3	contents:html="<html>..."
	t2	contents:html="<html>..."
	t1	contents:html="<html>..."

列族anchor

行键	时间戳	列族anchor
"com.cnn.www"	t5	anchor:cnnsi.com=”CNN”
	t4	anchor:my.look.ca="CNN.com"

4.3.6面向列的存储

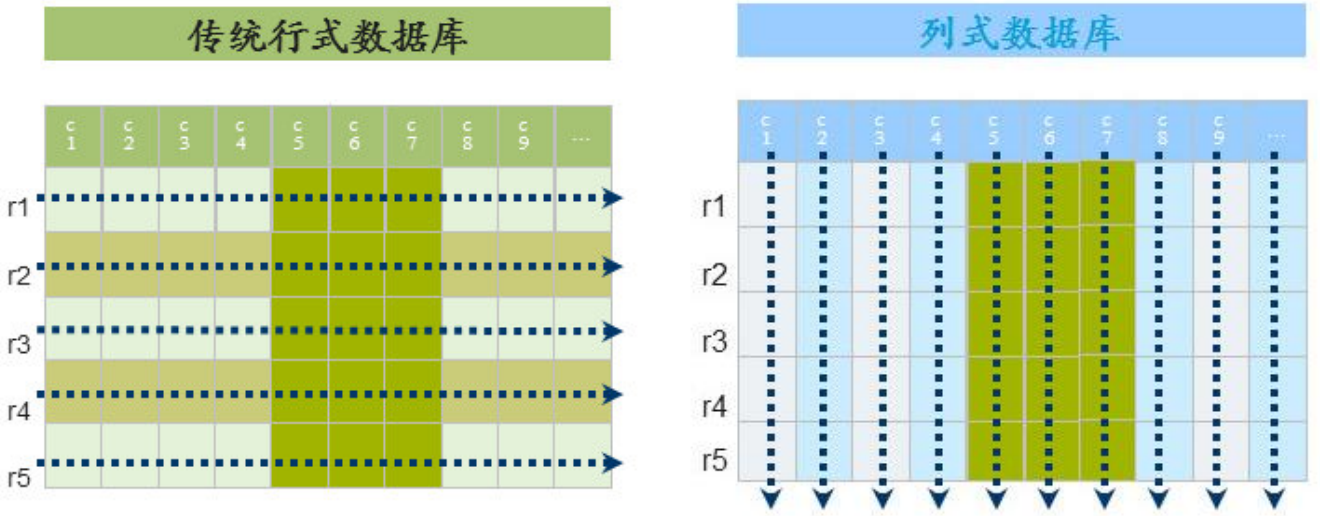


图4-3 行式数据库和列式数据库示意图

4.3.6面向列的存储

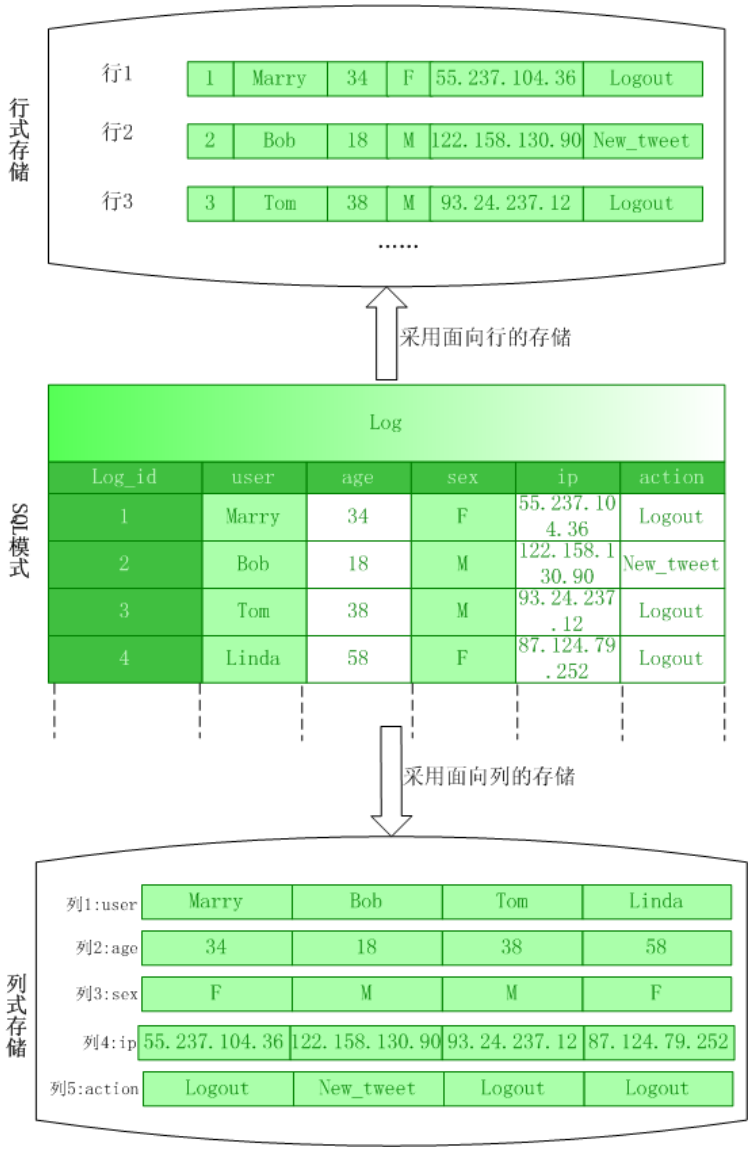


图4-4 行式存储结构和列式存储结构

4.4 HBase的实现原理

- 4.4.1 HBase功能组件
- 4.4.2 表和Region
- 4.4.3 Region的定位

4.4.1HBase功能组件

- HBase的实现包括三个主要的功能组件：
  - （1）库函数：链接到每个客户端
  - （2）一个Master主服务器
  - （3）许多个Region服务器
- Region服务器负责存储和维护分配给自己的Region，处理来自客户端的读写请求
- 主服务器Master负责管理和维护HBase表的分区信息
- 客户端并不是直接从Master主服务器上读取数据，而是在获得Region的存储位置信息后，直接从Region服务器上读取数据

4.4.2表和Region

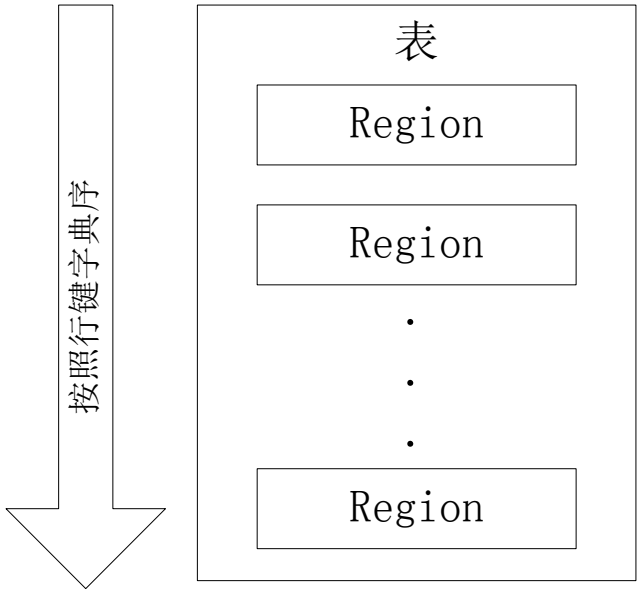


图4-5一个HBase表被划分成多个Region

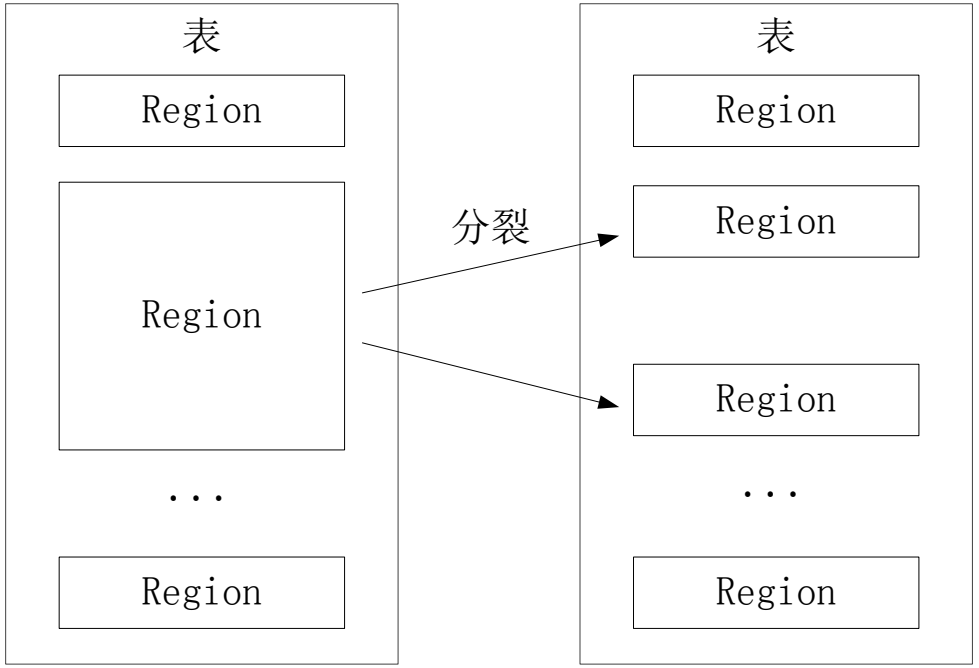


图4-6 一个Region会分裂成多个新的Region

4.4.2表和Region

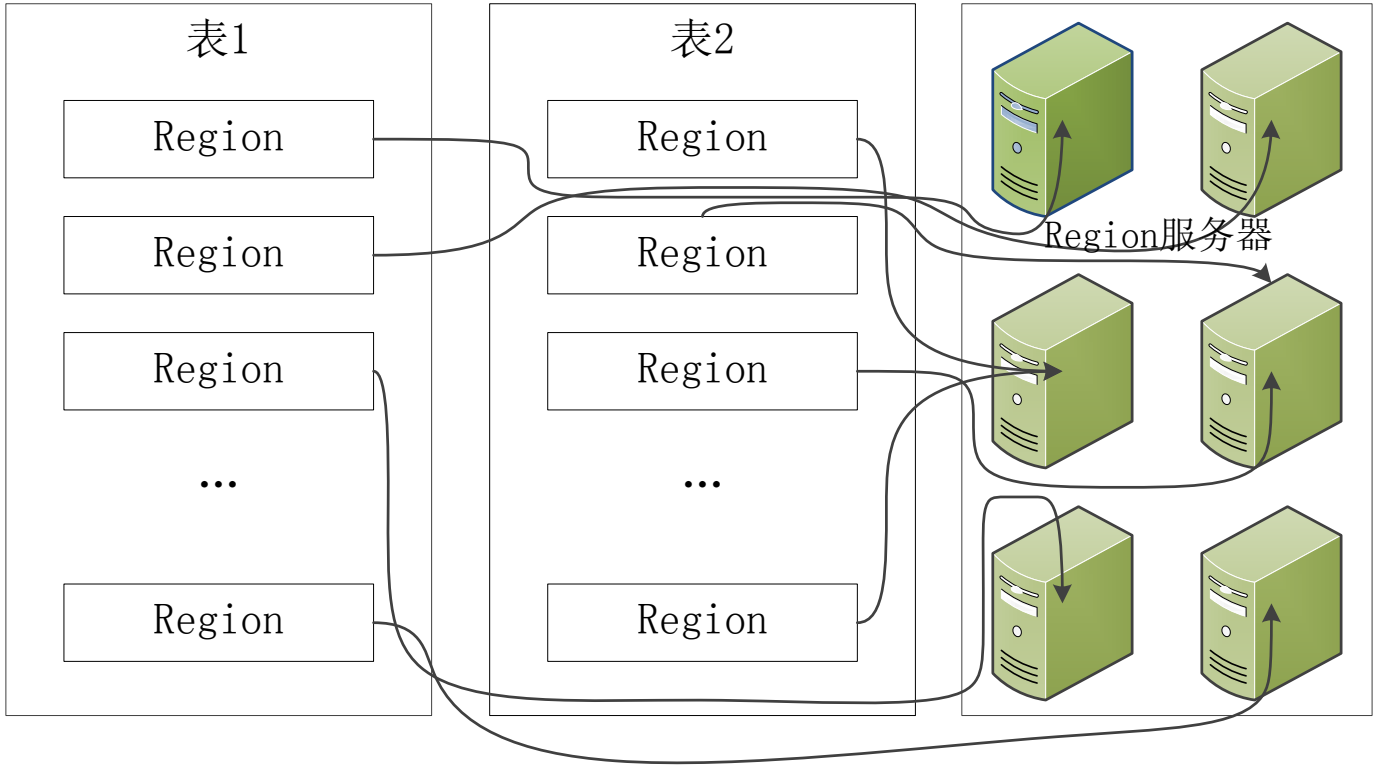


图4-7 不同的Region可以分布在不同的Region服务器上

4.4.3Region的定位

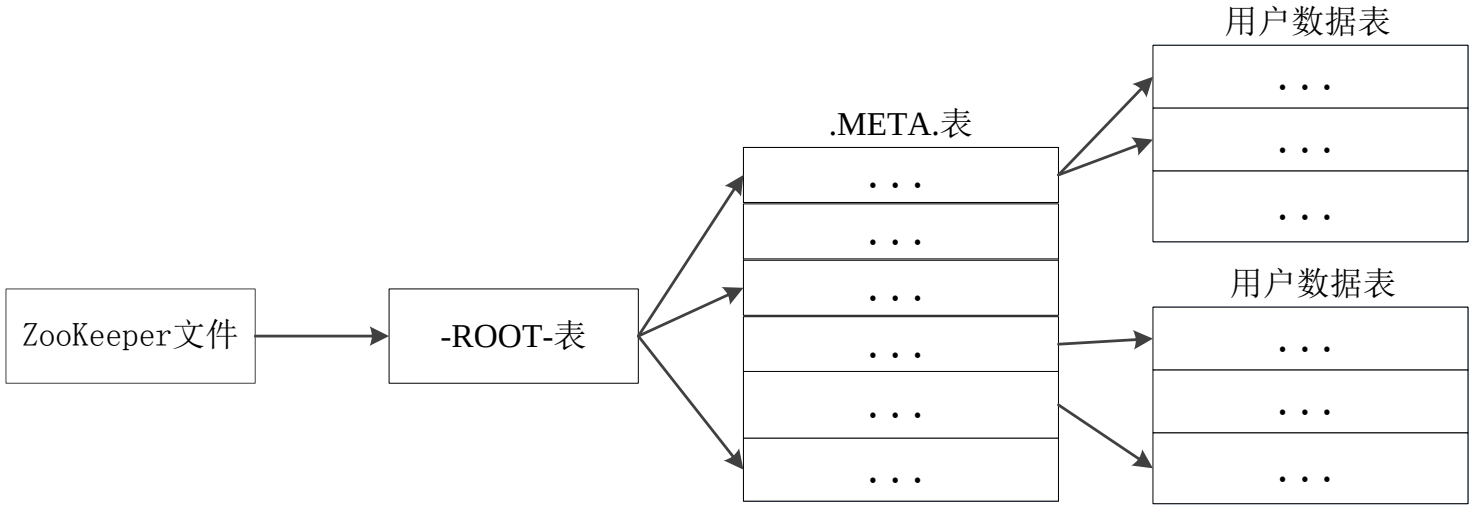


图4-8 HBase的三层结构

4.4.3Region的定位

表4-6 HBase的三层结构中各层次的名称和作用

层次	名称	作用
第一层	Zookeeper文件	记录了-ROOT-表的位置信息
第二层	-ROOT-表	记录了.META.表的Region位置信息 -ROOT-表只能有一个Region。通过-ROOT-表，就可以访问.META.表中的数据
第三层	.META.表	记录了用户数据表的Region位置信息， .META.表可以有多个Region，保存了HBase中所有用户数据表的Region位置信息

4.4.3Region的定位

•为了加快访问速度，.META.表的全部Region都会被保存在内存中
•假设.META.表的每行（一个映射条目）在内存中大约占用1KB，并且每个Region限制为128MB，那么，上面的三层结构可以保存的用户数据表的Region数目的计算方法是：
•（-ROOT-表能够寻址的.META.表的Region个数）×（每个.META.表的Region可以寻址的用户数据表的Region个数）
•一个-ROOT-表最多只能有一个Region，也就是最多只能有128MB，按照每行（一个映射条目）占用1KB内存计算，128MB空间可以容纳128MB/1KB=2^{17}行，也就是说，一个-ROOT-表可以寻址2^{17}个.META.表的Region。
•同理，每个.META.表的 Region可以寻址的用户数据表的Region个数是128MB/1KB=2^{17}。
•最终，三层结构可以保存的Region数目是(128MB/1KB) × (128MB/1KB) = 2^{34}个Region

4.5 HBase运行机制

- 4.5.1 HBase系统架构
- 4.5.2 Region服务器工作原理
- 4.5.3 Store工作原理
- 4.5.4 HLog工作原理

4.5.1HBase系统架构

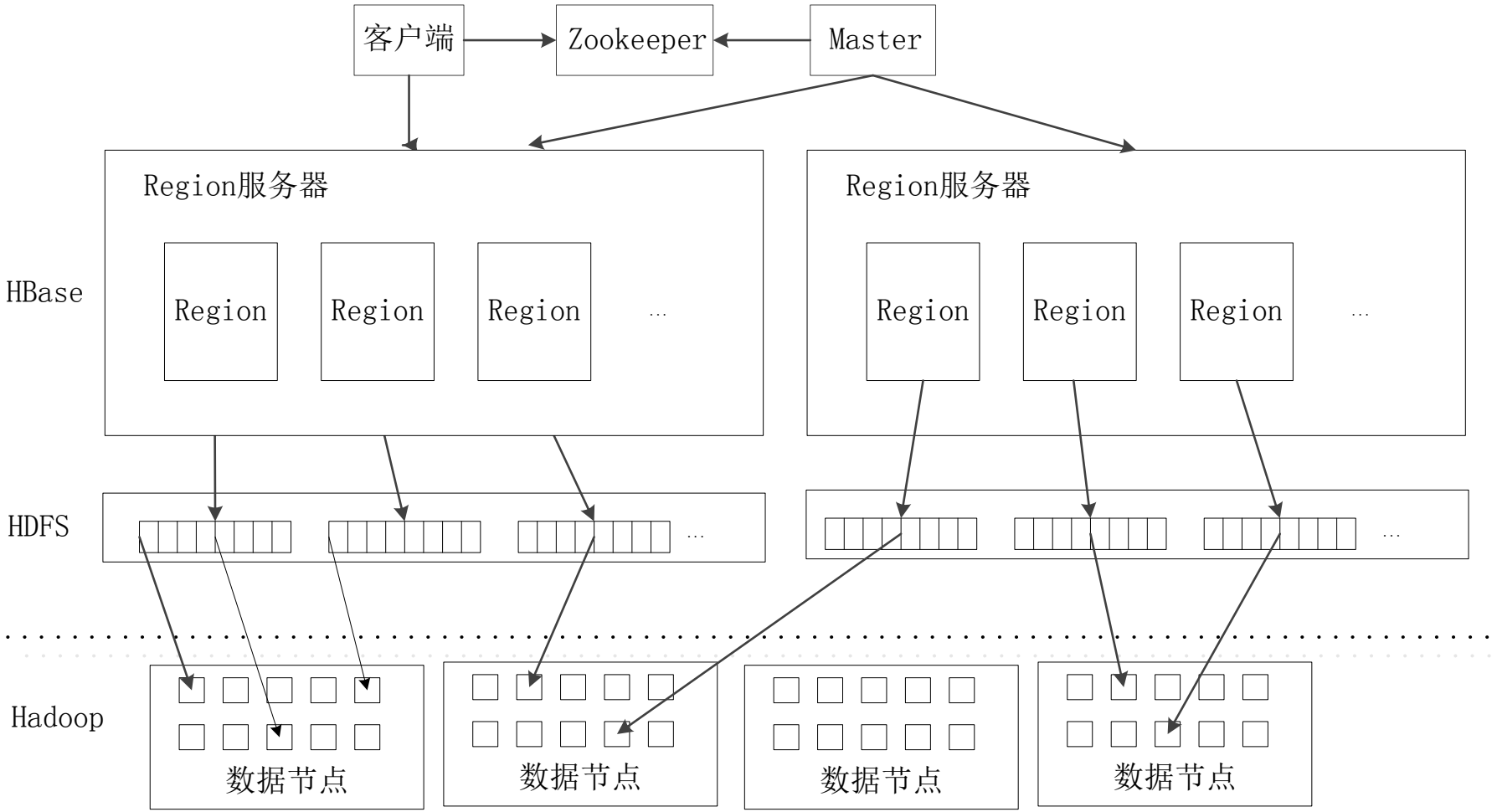
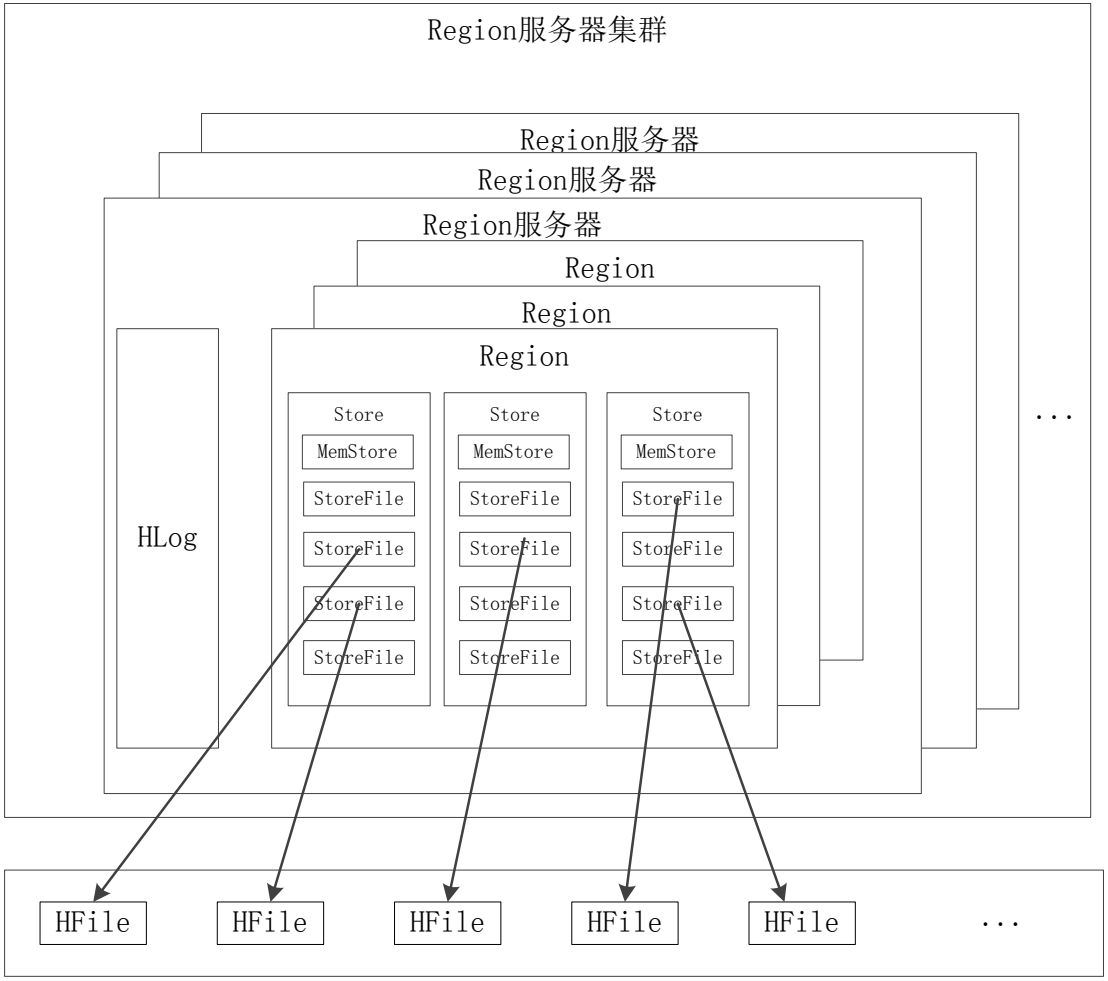


图4-9 HBase的系统架构

4.5.1HBase系统架构

- 1. 客户端
  - 客户端包含访问HBase的接口，同时在缓存中维护着已经访问过的Region位置信息，用来加快后续数据访问过程
- 2. Zookeeper服务器
  - Master可以通过Zookeeper随时感知各个Region服务器的工作状态
  - Zookeeper可以帮助选举出一个Master作为集群的总管，并保证在任何时刻总有唯一一个Master在运行，这就避免了Master的“单点失效”问题
  - Zookeeper保存了ROOT表地址
- 3. Master
- 主服务器Master主要负责表和Region的管理工作：
  - 管理用户对表的增加、删除、修改、查询等操作
  - 实现不同Region服务器之间的负载均衡
  - 在Region分裂或合并后，负责重新调整Region的分布
  - 对发生故障失效的Region服务器上的Region进行迁移
- 4. Region服务器
  - Region服务器是HBase中最核心的模块，负责维护分配给自己的Region，并响应用户的读写请求

4.5.2Region服务器工作原理



1. 用户读写数据过程
2. 缓存的刷新
3. StoreFile的合并

图4-10 Region服务器向HDFS文件系统中读写数据

4.5.3Store工作原理

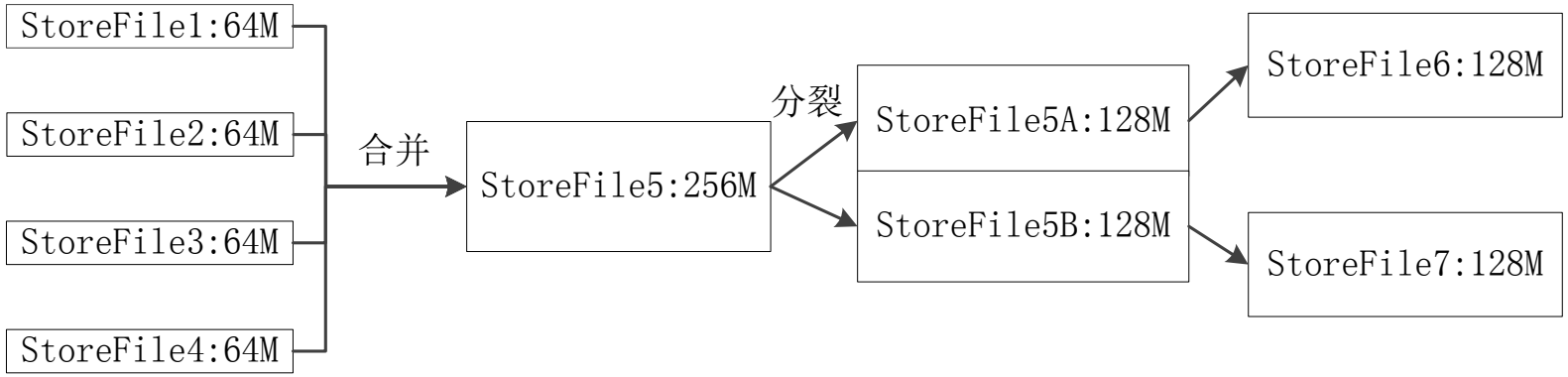


图4-11 StoreFile的合并和分裂过程

4.5.4HLog工作原理

- HBase系统为每个Region服务器配置了一个HLog文件，它是一种预写式日志（Write Ahead Log）
- Zookeeper会实时监测每个Region服务器的状态，当某个Region服务器发生故障时，Zookeeper会通知Master
- Master首先会处理该故障Region服务器上面遗留的HLog文件，这个遗留的HLog文件中包含了来自多个Region对象的日志记录
- 系统会根据每条日志记录所属的Region对象对HLog数据进行拆分，分别放到相应Region对象的目录下，然后，再将失效的Region重新分配到可用的Region服务器中，并把与该Region对象相关的HLog日志记录也发送给相应的Region服务器
- Region服务器领取到分配给自己的Region对象以及与之相关的HLog日志记录以后，会重新做一遍日志记录中的各种操作，把日志记录中的数据写入到MemStore缓存中，然后，刷新到磁盘的StoreFile文件中，完成数据恢复
- 共用日志优点：提高对表的写操作性能；缺点：恢复时需要分拆日志

本章小结

- 本章详细介绍了HBase数据库的知识。HBase数据库是BigTable的开源实现，和BigTable一样，支持大规模海量数据，分布式并发数据处理效率极高，易于扩展且支持动态伸缩，适用于廉价设备
- HBase可以支持Native Java API、HBase Shell、Thrift Gateway、REST Gateway、Pig、Hive等多种访问接口，可以根据具体应用场合选择相应访问方式
- HBase实际上就是一个稀疏、多维、持久化存储的映射表，它采用行键、列键和时间戳进行索引，每个值都是未经解释的字符串。本章介绍了HBase数据在概念视图和物理视图中的差别
- HBase采用分区存储，一个大的表会被分拆许多个Region，这些Region会被分发到不同的服务器上实现分布式存储
- HBase的系统架构包括客户端、Zookeeper服务器、Master主服务器、Region服务器。客户端包含访问HBase的接口；Zookeeper服务器负责提供稳定可靠的协同服务；Master主服务器主要负责表和Region的管理工作；Region服务器负责维护分配给自己的Region，并响应用户的读写请求

主讲教师

主讲教师：林子雨

单位：厦门大学计算机科学系
E-mail: ziyulin@xmu.edu.cn
个人网页：http://www.cs.xmu.edu.cn/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），现为厦门大学计算机科学系助理教授（讲师），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年度厦门大学奖教金获得者。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，编著出版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》并成为畅销书籍，编著并免费网络发布40余万字中国高校第一本闪存数据库研究专著《闪存数据库概念与技术》；主讲厦门大学计算机系本科生课程《数据库系统原理》和研究生课程《分布式数据库》《大数据技术基础》。具有丰富的政府和企业信息化培训经验，曾先后给中国移动通信集团公司、福州马尾区政府、福建省物联网科学研究院、石狮市物流协会、厦门市物流协会等多家单位和企业开展信息化培训，累计培训人数达2000人以上。

大数据学习教材推荐

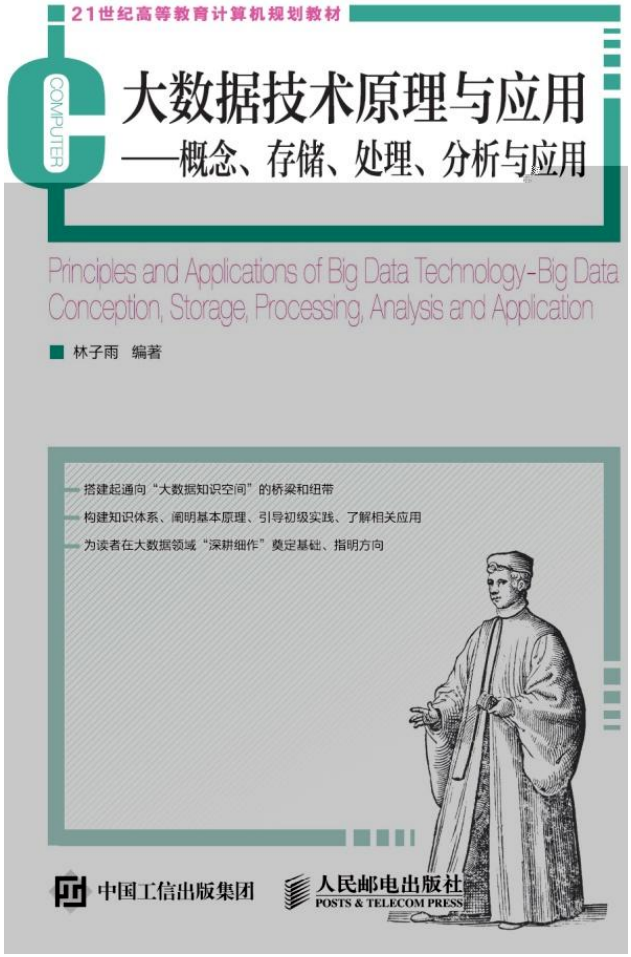
扫一扫访问教材官网

《大数据技术原理与应用——概念、存储、处理、分析与应用》，由厦门大学计算机科学系林子雨博士编著，是中国高校第一本系统介绍大数据知识的专业教材。

全书共有13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、流计算、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和MapReduce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/bigdata



课程建设单位

厦門大學数据库实验室
Database Lab of Xiamen University

厦門大學 云计算与大数据研究中心
XIAMEN UNIVERSITY Center for Cloud Computing and Big Data

海峡云计算与大数据应用研究中心
Strait Cloud Computing and Big Data Application Research Center


Thank You!