我们将从以下5个部分来介绍HBASE，第1部分是简介，第2部分是HBASE数据模型，第3部分是HBASE的实现原理，第4部分是HBASE运行机制，第5部分是HBASE的应用方案。

先来看第一部分，涉及两个内容，一是HBase简介，另一个是HBase与传统关系数据库的对比分析。

Hbase是Google BigTable的开源实现，是面向列、可伸缩的分布式数据库，主要用存储非结构化和半结构化的松散数据，能够处理非常庞大的表，达到超过10亿行数据和数百万列元素的数据表。这张图给出了Hbase在整个Hadoop生态系统中的位置。Hbase的底层通过HDFS来进行存储数据，MapReduce可以对Hbase的相关数据进行处理，上层的Pig、Hive等也可访问Hbase

中的数据。

HBase和BigTable的区别是什么呢？我们来看看这张表，主要从文件存储系统、海量数据处理、协同服务管理来比较两者不同，首先从文件存储系统来看，BigTable的底层是GFS存储系统，而Hbase是采用HDFS系统来存储，对于海量数据处理方式，BigTable用的是MapReduce,而Hbase用的是Hadoop MapReduce。对于协同服务管理，BigTable是用Chubby

来管理的，Hbase是用一个很好的、很有名的管理工具Zookeeper来进行管理的，主要用于集群服务的管理。

下面就要回答一个疑问，Hadoop已经有了HDFS和MapReduce，为什么需要HBase?其原因是，Hadoop MapReduce框架，它有一个特点就是高延迟数据处理的一个机制，没办法实现实时的处理应用的需求，同时hdfs他访问的一种方式是一种随批量访问模式，这不契合我们实际应用当中的一种需求，我们的应用要求实时随机访问数据。对传统的关心数据库，以前那种做法它是用分库的一种模式来实现一个系统的扩展性，但是这种分库都是人工手动的来实现的，而不是系统自己来维护的，后来也存在着一种分表的一种方式，能够人工的把这个表分在不同的服务器上，以减少访问量大而影响服务质量问题，最重要的一点，这种维护都必须要在停机的状态下才能进行，以致能改变数据结构的变化，同时数据结构中的空列存储空间也就浪费了，一旦空列数比较大的时候，这个存储空间浪费也就越大。所以我们在业界里面也出现了一项面向市场，面向现实应用的一类面向半结构化数据存储和处理的这样的高可扩展性的一个系统，比较典型的就是键值数据库和文档数据库，还有列族数据库，比如说是BigTable和Hbase。因此HBase的出现，已经为互联网服务领域和传统行业的众多在线式的数据分析处理系统解决了实际问题，所以也被众多行业所接纳和关注。

第2个方面就是HBase与传统数据库的对比，区别主要体现在以下六个方面，第1个方面就是数据类型的区别，关系型数据库采用关系模型，它有着丰富的数据类型，而HBase呢，采用的是比较简单的数据模型，它的存储直接用字符串方式，简单明了。第2个方面就是数据操作的区别，关系数据包含了丰富的操作，除了增删改查操作外，同时也包含了多表链接操作，比如说自然链接，等值链接，左链接，右链接等等，而HBase的操作就没有那么复杂了，都不会有多表的链接操作，只有一些简单的插入、查询、删除等等这些操作。这三个方面的区别就是存储模式的不同，关系数据库是基于行模式存储的，而HBase呢是基于列模式存储的，每个列都有几个文件保存，所以不同列族的文件也是分离的，这为大数据的一个存储和分析提供了一个很好的一个机制。

第4个方面是数据缩影的区别，关系型数据库是通过不同的列构建复杂的索引，通过索引以提高数据的访问性能，而HBase只有一个索引，就是行键，我们只要通过行键访问或者通过行键扫描就可以快速访问所有的数据，从而大大提高访问性能。第5个方面的区别是数据维护方面，在关系型数据库当中更新操作，会用一个新的值去替换记录当中原来的旧值，那么旧值就不会存在，而在HBase当中执行更新操作之后，不会删除以前的旧的值，而会生成一个新的版本的一个新值，不会替换旧值，旧值依然保留。第6个方面不同就是可申缩性，我们知道关系型数据库是很难实现横向扩展的，纵向扩展的空间也比较有限，而恰恰相反，HBase就可以很轻松的实现性能的伸缩，只需要在集群中增加或者减少硬件数量，来实现这个性能的伸缩。