



# 初识HBase

谭唐华

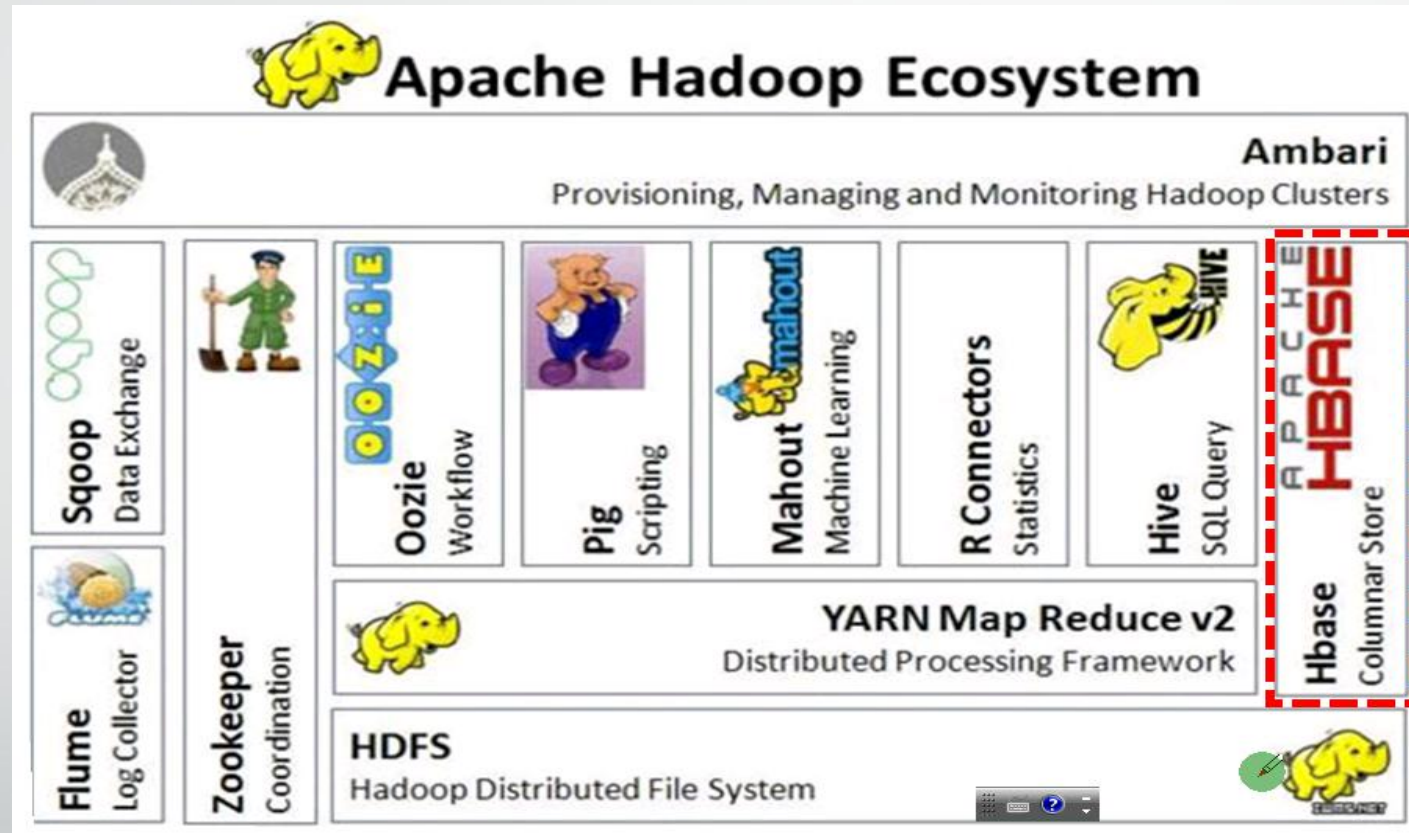
# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解

# 为什么使用HBase

- 传统的RDBMS关系型数据库（例如SQL）存储一定量数据时进行数据检索没有问题，可当数据量上升到非常巨大规模的数据（TB或PB）级别时，传统的RDBMS已无法支撑，这时候就需要一种新型的数据库系统更好更快的处理这些数据。我们可以选择HBase。

# HBase的地位



HBase占有举足轻重的作用，它居于HDFS之上，与MapReduce可以集成，与Hive也是集成的，HBase表中的数据与Hive表的数据可以关联，而后面的Spark也可以读HBase的数据。

# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解

# HBase是什么



- HBase技术来源于 Fay Chang 所撰写的Google论文 “Bigtable：一个结构化数据的分布式存储系统”。
- HBase是Apache的Hadoop项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。
- HBase – Hadoop Database，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统，利用HBase技术可在廉价PC Server上搭建起大规模结构化存储集群。
- HBase实际上是一个Hadoop的数据库系统，它的主要作用 and 传统数据库系统一样存储数据和检索数据。不同的是，HBase可以存储海量数据及海量数据的检索。

# HBase与Hadoop的对比

Hadoop/HDFS	HBase
为分布式存储提供文件系统	提供表状的面向列的数据存储
针对存储大尺寸的文件进行优化，不需要对这些文件进行随机读写	针对表状数据的随机读写进行优化
直接使用文件	使用key-value对数据
数据模型不灵活	提供灵活的数据模型
使用文件系统和处理框架	使用表状存储，依赖内置的Hadoop MapReduce支持
为一次写多次读进行优化	为多次读写进行优化

# HBase与关系型数据库的功能对比

关系型数据库RDBMS	HBase
支持向上扩展。（若需要更多的磁盘、内存和处理能力，需要升级服务器）	支持向外扩展。（若需要更多的磁盘、内存和处理能力，不需要升级服务器，需要为集群添加新的服务器）
使用SQL查询从表中读取数据	使用API和MapReduce来访问HBase表的数据
面向行（每行数据都是一个连续的页的单元）	面向列（每列数据都是一个连续的页的单元）
数据总量依赖于服务器配置	数据总量不依赖于服务器配置，而是总的机器数量
模式更严格	模式灵活，不太严格
具有ACID支持	没有内建的对HBase的支持
适合结构化数据	适合结构化和非结构化数据
传统关系型数据库一般是中心化的	通常是分布式的
一般能保证事务完整性	HBase不支持事务
支持JOIN	不支持JOIN
支持参照完整性	没有内置的参照完整性支持



# HBase与关系型数据库的功能对比

- 当数据量比较小，RDBMS可以支撑的时候，可以用RDBMS来实现，若需要在线事务处理时，RDBMS是合适的。
- 但当有海量数据需要处理时，可以选择HBase，并且HBase由于是列存储型数据库，在聚合计算和数据分析时非常快。

列存储数据库的优点	列存储数据库的缺点
具有高效和数据压缩的内部支持	JOIN和多表合并数据的查询性能不好
支持快速数据检索	更新过程中有大量的写入和删除操作，需要频繁合并和分裂，降低了存储效率
管理和配置简单。支持横向扩展	对关系模型支持不好，分区和索引模式设计比较困难
聚合查询的性能非常高	
可高效地进行分区	

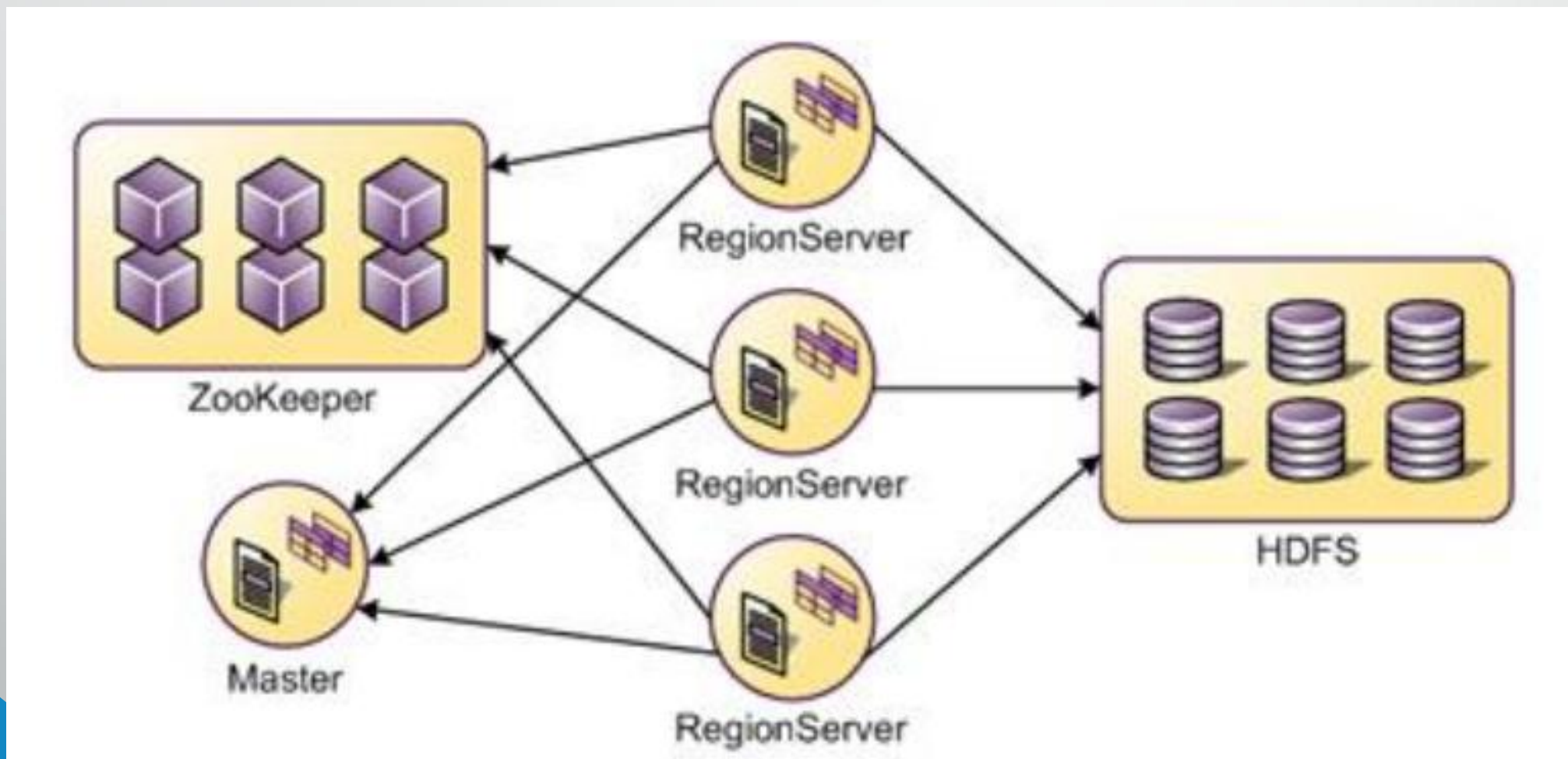
# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解

# HBase架构设计及表的存储设计

- HBase是水平扩展的、分布式的、开源有序映射数据库。
- 它运行在Hadoop文件系统HDFS上。它不要求有预定义的模式，可以被看做弹性扩展的多维表格，通过动态添加列，在数据插入或查询之前修改列结构，以支持任意的数据结构。
- HBase是一个建立在HDFS上的列存储数据库，具有至此线性扩展（横向扩展）、自动故障转移、自动分区及模式自由等特性。

# HBase体系结构图



# HBase架构设计及表的存储设计

## ◆ Master :

- 为HBase的主节点，用来协调客户端应用程序和RegionServer的关系，同时用来监控和记录元数据的变化和管理。

## ◆ RegionServer :

- 是从节点，用region的形式处理实际的表。region是HBase表的基础单元组件，它存储了分布式表。所以HBase表和HBase集群利用Master和RegionServer来协同工作。

## ◆ ZooKeeper :

- 是一个高性能、集中化、分布式应用程序协调服务，它为HBase提供了分布式同步和组服务。在HBase中，它用来选举集群主节点Master，以便跟踪可用的在线服务器，同时维护集群的元数据。一般安装多个，用于提供Master的高可用性。

➤ 通常，Master和Hadoop的NameNode进程运行在同一台主机上，与DataNode通信以读写HDFS的数据。RegionServer跟Hadoop的DataNode运行在同一台主机上。

# HBase架构设计及表的存储设计

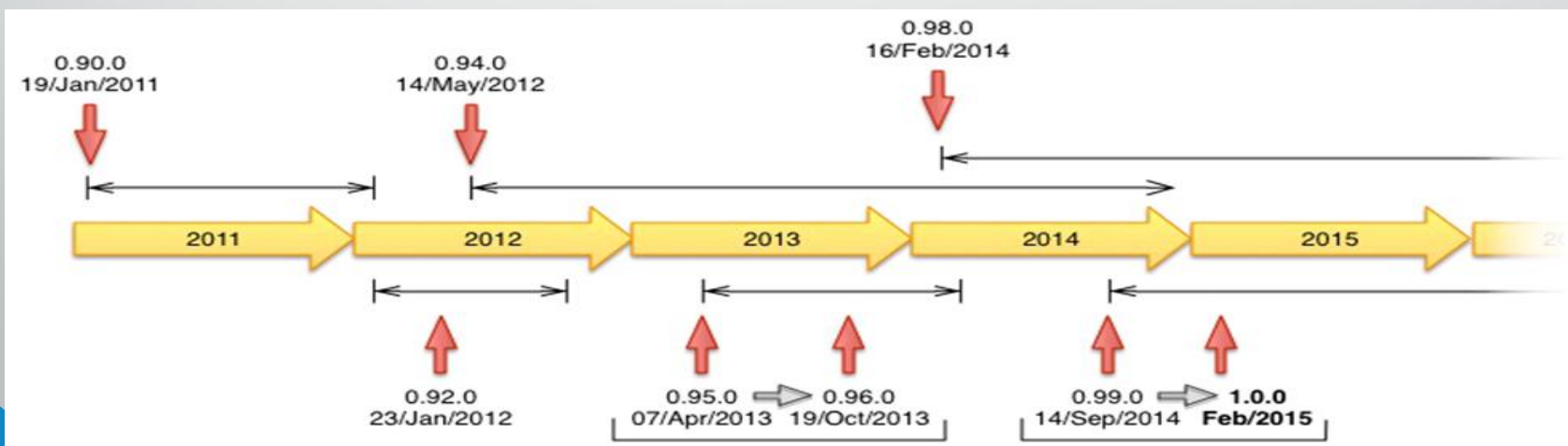
- **HBase具有扩展性主要原因在于：**
  - a. 数据在文件中，在HDFS中，DataNode可以任意加；
  - b. 一个表的数据分成很多个区域Region，每个区域均有一个RegionServer负责数据管理及检索，而RegionServer也可以任意加。它运行在DataNode的机器上，其数量可以跟DataNode一样多。

# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解

# HBase版本

- ◆ **2006年**：开始出现HBase的概念。
- ◆ **2007年**：以下功能被加入进HBase——在线添加或删除列族，启动或停止HBase集群的脚本，MapReduce连接器，HBase shell支持，支持行和列的过滤，分布式region算法，第一个rest接口。举办了第一届HBase聚会。
- ◆ **2008年**：HBase从0.1.0升级到0.18.1，HBase作为Hadoop的贡献者，称为Hadoop的子项目，第一次独立发布。
- ◆ **2009年**：HBase从0.19.0升级到0.20.\*，在数据读写、批量数据读写、块压缩、REST接口上都有改进，添加了二进制比较器和正则表达式过滤器等。
- ◆ **2010年到现在**：HBase从0.89.\*升级到1.0.0，增加了以下功能——依赖Hadoop HA的HBase高可用性，轮流升级，故障检测和恢复的改进，单元格层次的安全访问，反向扫描等功能，如下图：



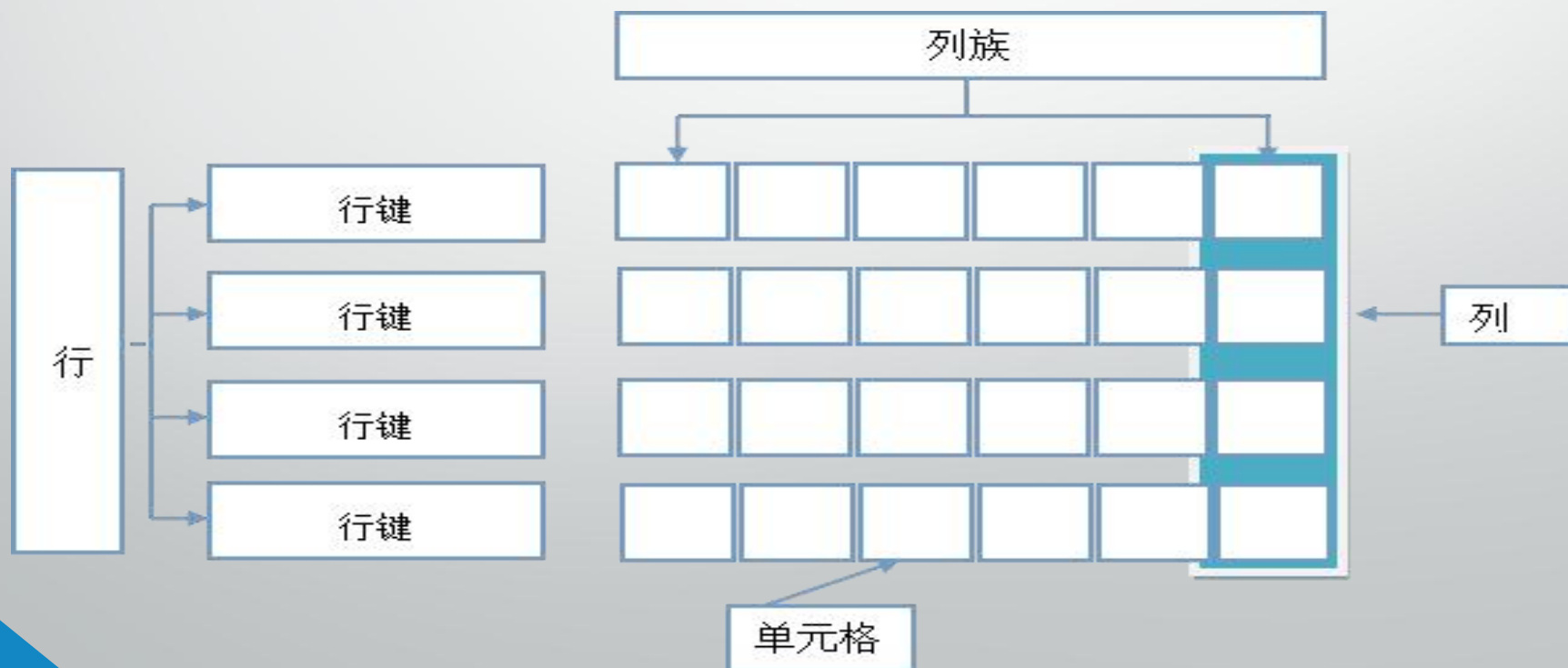


# HBase数据存储模型

- ◆ HBase的数据类型
- ◆ HBase数据模型术语
- ◆ HBase中表的数据模型术语举例
- ◆ HBase中表的基本概念

# HBase数据存储模型

- HBase不是以关系设计为中心，它是根据用户需求更灵活的开放设计。它提供了在行键上的单一索引，这在关系世界里称为主键。我们可以通过把行划分为列族和列来避免大的读取和写入操作，并且这种方式支持水平切分和垂直切分。
- 一个HBase表由以下几部分组成：



# HBase的数据类型

- HBase中没有花哨的数据类型，它所有都是字节数组。它是一种字节进字节出的数据库，其特征不在于，当插入一个值时，HBase隐式地通过序列化框架将数据转换成字节数组，然后存储进单元格，或者给出字节数组。
- 当添加或者获取数值时，它隐式地转换成等价的数据展示出来。
- HBase的单元格只能容纳字节数组。任何可以转换成字节的数据都可以存储在HBase中。
- 可以存储10-15MB的值到HBase的单元格中，但如果值太大，可以将文件存储到HDFS中，然后在HBase中存储文件的路径。
- 不建议将一个巨大的文件或值转换成字节数组存储在HBase中；但是HDFS用到的主机文件和文件元数据可以存储到HBase中。

# HBase数据模型术语

- **行键**

- 这是HBase表中每个记录的唯一键，无论选择什么类型数据作为行键，它在内部、磁盘或内存里，都将转换为字节数组进行存储。
- 表中的每条数据有唯一的标识符，即rowkey，类似于关系型数据库的主键。

- **列族**

- 一张HBase表由表的不同列集合在一起。将相同功能或类型的列分类组合在一起，这样做的好处是可以更快的分开存储在HBase磁盘上的列族中检索出所需的列。

- **列**

- 列属于某一个column family列族。

# HBase数据模型术语

- **版本**

- HBase能够为一个单元格元组（行、列族和列）保存多个值，每个单元格被称为一个记录的版本。版本制定为基于时间戳的长整形。默认情况下，HBase保留3个版本的记录。当然，我们也可以改变保留版本的数量，我们也可以通过指定来获取某个特定的版本。

- **时间戳**

- 对于每个插入的数据，当前的时间戳与值是相关的，它表示了数值插入到表中的时间。

- **单元格**

- 最小或基本的存储单元，在内部是一个列的实际值存储。故插单元格数据时必须包含rowkey+ columnfamily（列族名）+columnname（列名）+timestamp：value。

# HBase中表的数据模型术语举例

- 以员工employee表为例，employee具有以下特征basic（基本信息）、job（工作信息）、edu（教育信息），表格如下：

employee（员工的信息描述）		
basic（基本信息）	job（工作信息）	edu（教育信息）
IDCard（身份证号码）	job1（工作经历1）	edu 1（教育信息1）
name（姓名）	job2（工作经历2）	edu 2（教育信息2）
age（年龄）		
birthday（生日）		
sex（性别）		

## ■ 以上表中各数据模型术语对照如下：

- 行键：IDCard（身份证号码）
- 列族：basic（基本信息）、job（工作信息）、edu（教育信息）
- 列：name（姓名）、job1（工作经历1）、edu 1（教育信息1）等
- 单元格：举例插入张三员工的名称，表示为：IDCard（身份证号码）+ basic+name+20160130：张三

# HBase中表的基本概念

Table in HBase

- Schema:TableName & Column Family Name ;
- value:is stored in column with version as byte[ ];

- ◆ HBase的Schema由表名和列族两项组成。另外值结合版本信息转为字节数组存储在列中。

Column Family		Column Family		
Column Label	Column Qualifier	CL	CL	CL

- ◆ 下图中zhangsan、wangwu是rowkey ; name和contact是列族 ; firstname、lastname、nickname和email、phone是列名 ; san、zhang等其他单元格就是值了。

	name			contact	
	<u>firstname</u>	<u>lastname</u>	nickname	email	phone
<u>zhangsan</u>	san	<u>zhang</u>	bluedavy	<u>**@gmail</u>	186*****
<u>wangwu</u>	<u>wu</u>	<u>wang</u>			158*****

# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解



# HBase部署架构及安装部署启动讲解

➤ 下面让我们来配置HBase，HBase有以下几种运行模式。

- **单机**：此模式使用本地文件系统而不是HDFS存储表和数据，所有的守护进程运行在一个JVM中。此模式一般用于测试，如果你想要展现HBase的真正威力，还需要结合Hadoop。
- **分布式**：分布式分为两种模式。
  - **伪分布**：使用HDFS文件系统，所有守护进程运行在一个JVM中，用于测试目的。可以利用到Hadoop的威力，进行少量的配置。
  - **完全分布**：使用HDFS作为底层文件系统，所有的守护进程运行在不同机器上的JVM中，适用于生产环境。

# 在单机模式中配置HBase

配置单机模式非常简单，只需要下载TAR包、解压缩，然后启动守护进程就可以。这种模式不会使用HDFS文件系统，而是使用本地文件系统file://来存储数据。

## ◆ 步骤一：

- 从<http://apache.mirrors.pair.com/hbase/> 上下载HBase二进制文件。从兼容表中可以找到HBase和Hadoop的兼容版本。

## ◆ 步骤二：

- 其次下载完成后，利用filezilla工具将TAR包放到/opt/software指定目录

# 在单机模式中配置HBase

- ◆ 步骤三：解压该tar包，利用以下命令将文件解压。
- ◆ 步骤四：利用以下命令启动namenode
- ◆ 步骤五：利用以下命令启动datanode。

◆ 步骤六：

1. 启用并修改hbase-env.sh配置文件中的JAVA\_HOME：找到配置文件中的JAVA\_HOME，后将JAVA\_HOME前的#号删除，修改其内容，结果为下图

```
export JAVA_HOME=/opt/modules/jdk1.7.0_67
```

2. 启用并修改hbase-env.sh配置文件中的HBASE\_MANAGES\_ZK：找到配置文件中的HBASE\_MANAGES\_ZK，后将JAVA\_HOME前的#号删除，修改其内容，结果为下图

```
export HBASE_MANAGES_ZK=false
```

或者该项不设置，表示告诉HBase，它是否需要管理自己的zoo\_keeper实例。

# 在单机模式中配置HBase

- 步骤七：利用notepad工具修改hbase-site.xml配置文件中的信息
- <property>
- <name>hbase.rootdir</name>
- <value>hdfs://hadoop-senior01.beifeng.com:8020/hbase</value>
- </property>
- <property>
- <name>hbase.cluster.distributed</name>
- <value>true</value>
- </property>
- <property>
- <name>hbase.zookeeper.quorum</name>
- <value>hadoop-senior01.beifeng.com</value>
- </property>

# 在单机模式中配置HBase

◆ 步骤八：修改regionservers文件，将内容修改为hostname本机名。

◆ 步骤九：替换lib目录关于hadoop和zookeeper的jar包

◆ 步骤十：启动hbase。命令如图：

注意：也可以分角色启动

```
bin/hbase-daemon.sh start master
```

```
bin/hbase-daemon.sh start regionserver
```

◆ web接口：

```
http://bigdata.beifeng.com:60010
```

# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解

# HBase Shell基本使用

shell	Run the HBase shell
hbck	Run the hbase 'fsck' tool
hlog	write-ahead-log analyzer
hfile	store file analyzer
zkcli	Run the ZooKeeper shell
upgrade	Upgrade hbase
master	Run an HBase HMaster node
regionserver	Run an HBase HRegionServer node
zookeeper	Run a Zookeeper server
rest	Run an HBase REST server
thrift	Run the HBase Thrift server
thrift2	Run the HBase Thrift2 server
clean	Run the HBase clean up script
classpath	Dump hbase CLASSPATH
mapredcp	Dump CLASSPATH entries required by mapreduce
pe	Run PerformanceEvaluation
ltt	Run LoadTestTool
version	Print the version
CLASSNAME	Run the class named CLASSNAME



# HBase Shell基本使用

- **\$ bin/hbase命令**
  - 查看hbase的版本
    - 语法格式: bin/hbase version
  - 连接HBase
    - 使用hbase 的shell命令连接正在运行的安装在你的HBase的bin目录中的HBase实例。语法格式: bin/hbase shell

# HBase常用命令

## 1. help命令

- 连接好HBase之后可使用help命令，回车后，屏幕上会显示HBase shell命令的一些基本用法，并有几个命令组。命令组 ( COMMAND GROUPS ) 包括：
  - 普通组：包含状态信息，表的帮助信息，帮助信息等方面的命令
  - ddl组：包括修改、创建、删除等数据定义语言方面的命令
  - dml组：包括添加、计数、删除、查询所有，获取部分信息等方面的数据操作语言方面的命令
  - namespace组：包括修改、创建、描述、删除命名空间等方面的命令
  - tools组：工具组命令
  - replication组命令
  - snapshots快照命令
  - security安全命令：包括授予、取消权限、用户许可方面的命令
  - visibility labels命令：包括添加label，清除、获取、设置认证等命令

- 以上命令中和关系型数据库语言一样，ddl、dml、security方面的命令是我们经常用的命令

# HBase常用命令

## 2. 创建一个表格

- 使用create命令来创建一个新的表格，你必须指明表名和列族名。
- 【注】可用help+‘命令名’ 查询该命令的用法。例如：help ‘create’
- 语法格式为：create ‘表名’ ; 列族名’
- 例题：create 'user','info'

## 3. 显示表格的信息

- 使用list命令来显示已创建好的所有表格信息或指定表格名的表格信息
- 语法格式如下：list 或list ‘表名’
- 例题：list ‘user’

# HBase常用命令

## 4.describe命令

- 利用describe显示表格描述信息
- 语法格式如下：describe '表名'
- 例题：describe 'user'

## 5.插入数据

- 使用put命令插入数据
- 语法如下：put '表名','行键','列族:列名','值'
- 例题如下：为表格user插入以下数据如表

# HBase常用命令

行键	列族	列名	值
10001	info	name	Zhangsan
		age	25
		sex	male
		address	shanghai
10002	info	name	Wangwu
		age	30
		qq	231294737
		tel	231294737

为创建以上表格,输入语句如下:

```
put 'user','10001','info:name','Zhangsan'
```

表示的含义是在user表中插入一个单元格值，键为用户编号10001，列族为info，列名为name，该列对于的值为Zhangsan

```
put 'user','10001','info:age','25'
```

同上例子，zhangsan性别sex为male

# HBase常用命令

③ `put 'user','10001','info:sex','male'`

■ 同上例子，Zhangsan性别sex为male

③ `put 'user','10001','info:address','shanghai'`

■ 同上Zhangsan地址address为上海

⑤ `put 'user','10002','info:name','Wangwu'`

⑥ `put 'user','10002','info:age','30'`

⑦ `put 'user','10002','info:qq','231294737'`

⑧ `put 'user','10002','info:tel','231294737'`

# HBase常用命令

## 6. HBase中数据查询方式

- ① 依据rowkey使用get命令查询，**语法格式**：**get** '表名' , 行键名' ,例：

```
hbase(main):017:0> get 'user','10001'
COLUMN                                CELL
info:address                          timestamp=1443301425881, value=shanghai
info:age                              timestamp=1443301392371, value=25
info:name                             timestamp=1443301377140, value=zhangsan
info:sex                              timestamp=1443301411897, value=male
```

- ② 语法格式：**get** '表名' , 行键名' , '列名'

```
hbase(main):018:0> get 'user','10001','info:name'
COLUMN                                CELL
info:name                             timestamp=1443301377140, value=zhangsan
1 row(s) in 0.0230 seconds
```



# HBase常用命令

```
hbase(main):023:0> scan 'user'
ROW COLUMN+CELL
10001 column=info:address, timestamp=1443301425881, value=shanghai
10001 column=info:age, timestamp=1443301392371, value=25
10001 column=info:name, timestamp=1443301377140, value=zhangsan
10001 column=info:sex, timestamp=1443301411897, value=male
10002 column=info:age, timestamp=1443301630415, value=30
10002 column=info:name, timestamp=1443301620248, value=wangwu
10002 column=info:qq, timestamp=1443301652155, value=231294737
10002 column=info:tel, timestamp=1443301639634, value=231294737
2 row(s) in 0.0610 seconds
```

- ③ 使用scan命令查询表格中某范围数据，语法格式如下：**scan '表名' , {COLUMN => [ '列族:列名,' '列族:列名' ]}**

在表中再插入一个用户信息，命令如下：**put 'user','10003','info:name','zhaoliu'**

```
hbase(main):027:0> scan 'user', {COLUMNS => ['info:name', 'info:age']}
ROW COLUMN+CELL
10001 column=info:age, timestamp=1443301392371, value=25
10001 column=info:name, timestamp=1443301377140, value=zhangsan
10002 column=info:age, timestamp=1443301630415, value=30
10002 column=info:name, timestamp=1443301620248, value=wangwu
10003 column=info:name, timestamp=1443301696356, value=zhaoliu
3 row(s) in 0.0390 seconds
```



# 课程大纲

- ◆ HBase是什么?
- ◆ 理解HBase
- ◆ HBase架构设计及表的存储设计
- ◆ HBase版本及数据存储模型
- ◆ HBase部署架构及安装部署启动讲解
- ◆ HBase Shell基本使用（创建表，对数据的CRUD操作）
- ◆ HBase 表的物理模型讲解