CHTC
中国HBase技术社区

云栖社区

HBase基本知识介绍及典型案例分析

阿里云 吴阳平(明惠) 阿里云HBase业务架构师 过往记忆博主

目录 / Contents

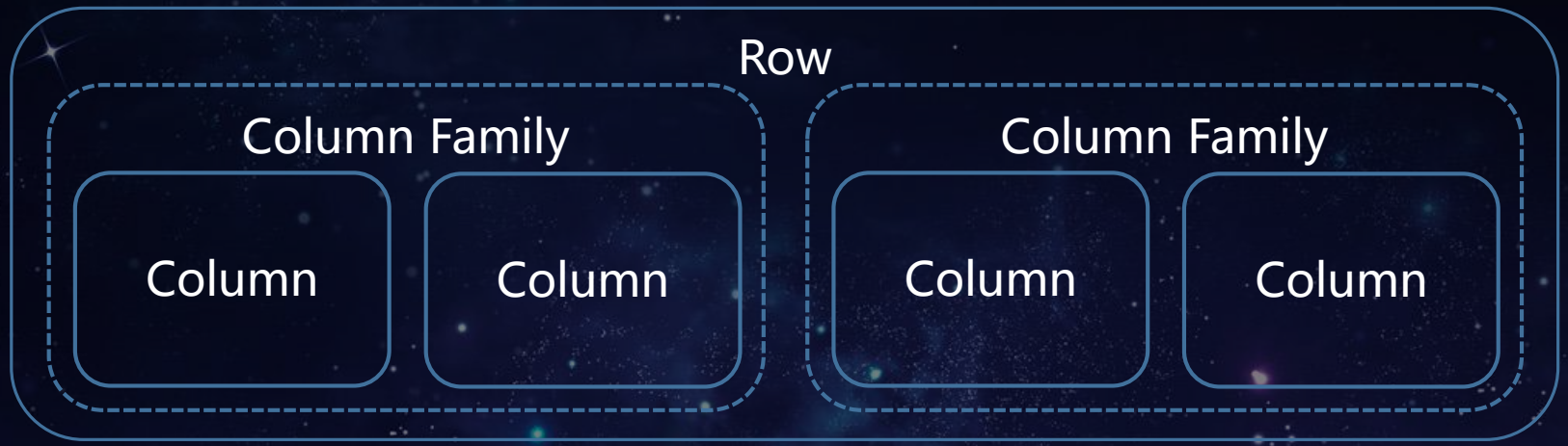
01 HBase快速入门

1 HBase是什么

- 分布式、多版本、面向列的开源KV数据库；
- 支持PB级、百万列的数据存储；
- 强一致性、高扩展、高可用。

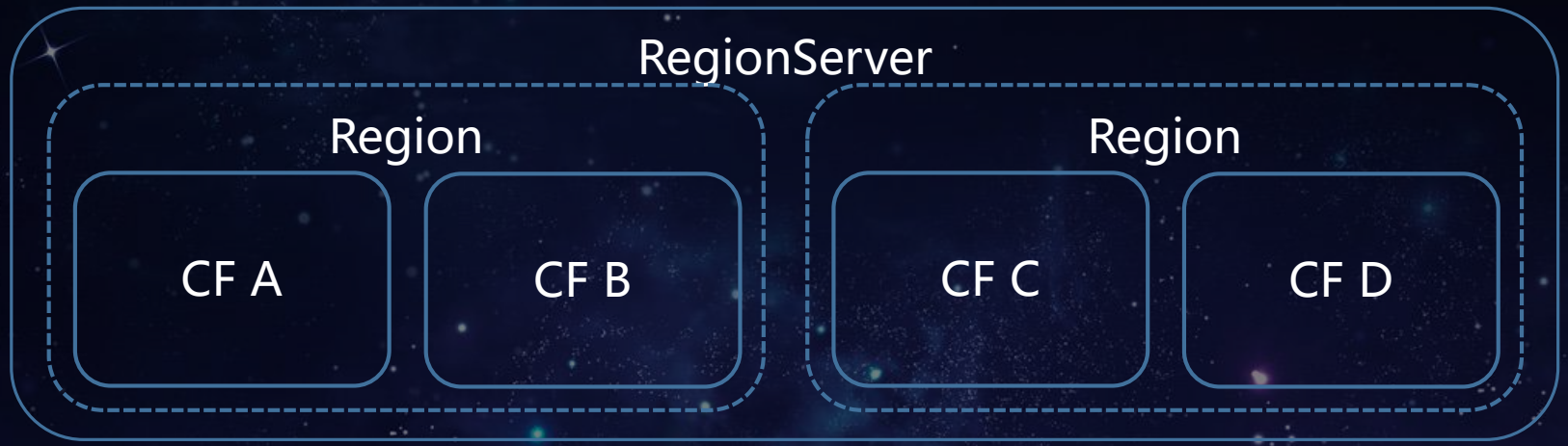
2 HBase表核心概念

- **RowKey**：表中每条记录的主键；
- **Column Family**：列族，将表进行横向切割，后面简称CF；
- **Column**：属于某一个列族，可动态添加列；
- **Version Number**：类型为Long，默认值是系统时间戳，可由用户自定义；
- **Value**：真实的数据。



3 HBase表核心概念

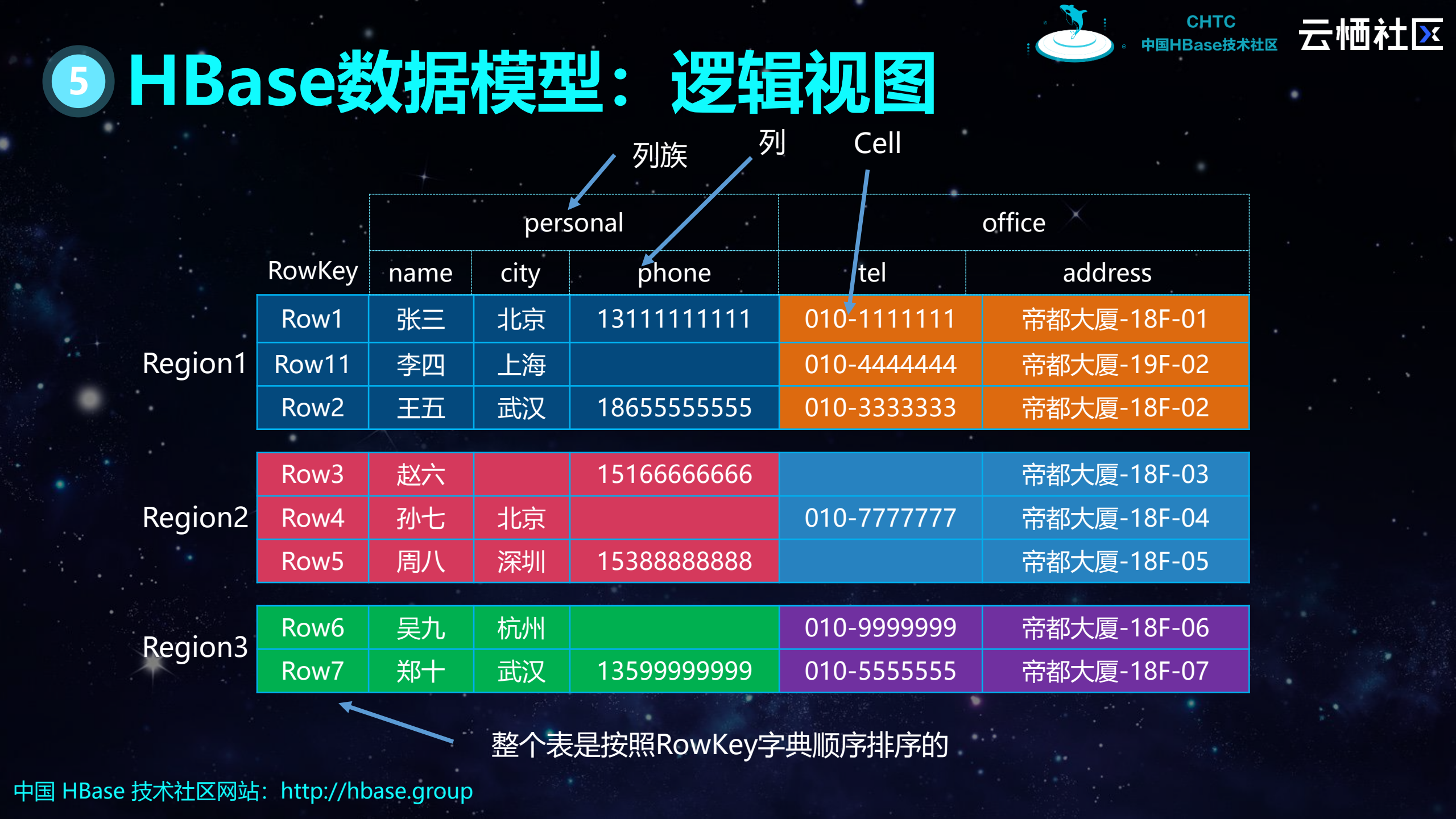
- **Region**：一段数据的集合；
- **RegionServer**：用于存放Region的服务。



4 HBase数据模型：逻辑视图

Row1	张三	北京	13111111111	010-1111111	帝都大厦-18F-01
Row11	李四	上海		010-4444444	帝都大厦-19F-02
Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Row3	赵六		15166666666		帝都大厦-18F-03
Row4	孙七	北京		010-7777777	帝都大厦-18F-04
Row5	周八	深圳	15388888888		帝都大厦-18F-05
Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

5 HBase数据模型：逻辑视图



	RowKey	personal			office	
		name	city	phone	tel	address
Region1	Row1	张三	北京	13111111111	010-1111111	帝都大厦-18F-01
	Row11	李四	上海		010-4444444	帝都大厦-19F-02
	Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Region2	Row3	赵六		15166666666		帝都大厦-18F-03
	Row4	孙七	北京		010-7777777	帝都大厦-18F-04
	Row5	周八	深圳	15388888888		帝都大厦-18F-05
Region3	Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
	Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

整个表是按照RowKey字典顺序排序的

6 HBase数据模型：物理视图

Key

Key Length	Value Length	Row Length	Row Key	CF Length	CF	Column Qualifier	Time Stamp	Key Type	Value

Row Key	CF	Column Qualifier	Time Stamp	Value

Row Key	CF	Column Qualifier	Time Stamp	Value
Row1	personal	name	1539684094	张三

7 HBase数据模型：物理视图

RowKey	personal			office	
	name	city	phone	tel	address
Row1	张三	北京	13111111111	010-1111111	帝都大厦-18F-01
Row11	李四	上海		010-4444444	帝都大厦-19F-02
Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Row3	赵六		15166666666		帝都大厦-18F-03
Row4	孙七	北京		010-7777777	帝都大厦-18F-04
Row5	周八	深圳	15388888888		帝都大厦-18F-05
Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

- 数据是以 KV 形式存储；
- 每个 KV 只存储一个cell里面的数据；
- 不同CF的数据是存在不同的文件里面。

Row Key	CF	CQ	Time Stamp	Value
Row1	personal	name	1539684094	张三
Row1	personal	city	1539684095	北京
Row1	personal	phone	1539684096	13111111111
Row11	personal	name	1539684094	李四
Row11	personal	city	1539684093	上海
Row2	personal	name	1539684092	王五

Row Key	CF	CQ	Time Stamp	Value
Row1	office	tel	1539684043	010-1111111
Row1	office	address	1539684095	帝都大厦-18F-01
Row11	office	tel	1539684096	010-4444444
Row11	office	address	1539684094	帝都大厦-19F-02
Row2	office	tel	1539684093	010-3333333
Row2	office	address	1539684092	帝都大厦-18F-02

8 HBase数据模型：物理视图

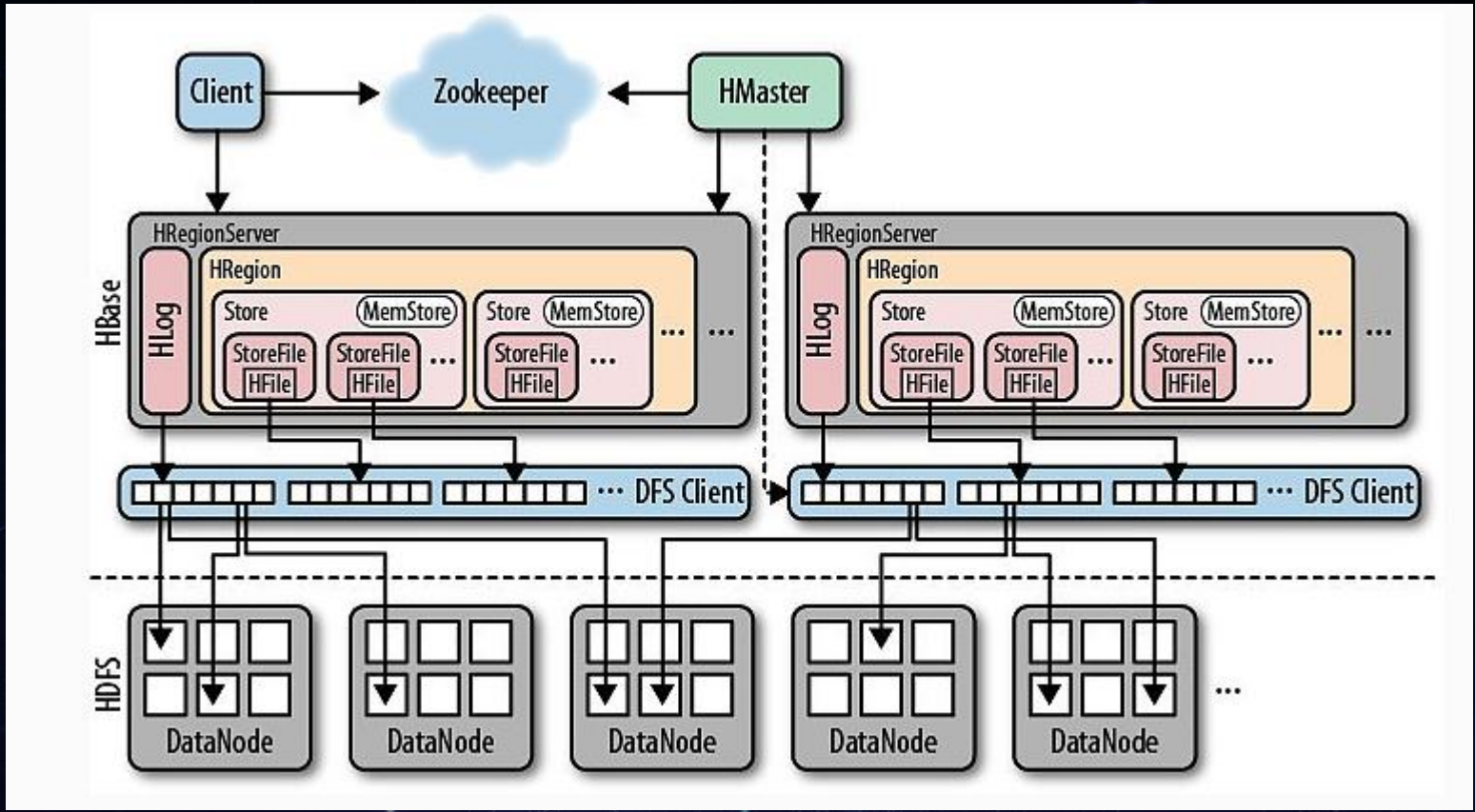
RowKey	personal			office	
	name	city	phone	tel	address
Row1	张三	上海	13111111111	010-1111111	帝都大厦-18F-01
Row11	李四	上海		010-4444444	帝都大厦-19F-02
Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Row3	赵六		15166666666		帝都大厦-18F-03
Row4	孙七	北京		010-7777777	帝都大厦-18F-04
Row5	周八	深圳	15388888888		帝都大厦-18F-05
Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

- HBase支持数据多版本特性，通过带有不同时间戳的多个KeyValue版本来实现的；
- 每次put，delete都会产生一个新的Cell，都拥有一个版本；
- 默认只存放数据的三个版本，可以配置；
- 查询默认返回最新版本的数据，可以通过制定版本号或版本数获取旧数据。

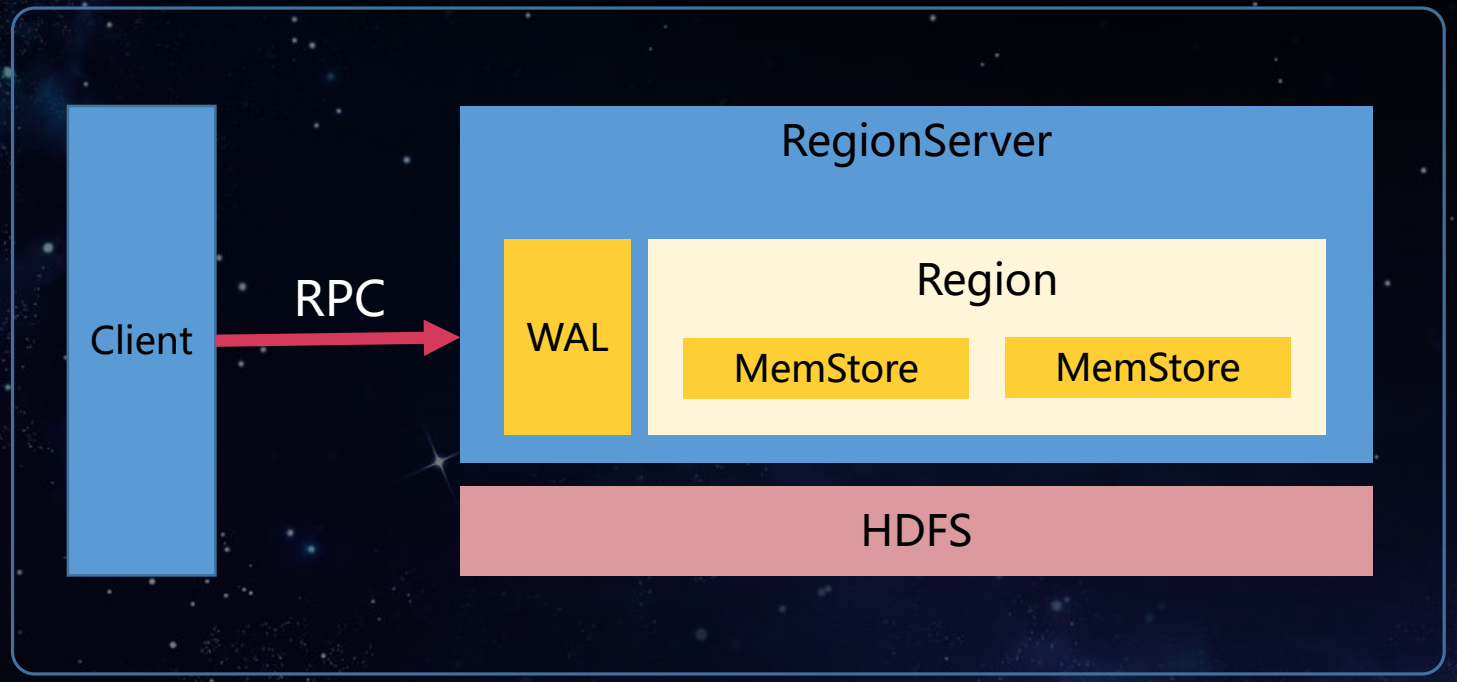
Row Key	CF	CQ	Time Stamp	Value
Row1	personal	name	1539684094	张三
Row1	personal	city	1539685089	上海
Row1	personal	city	1539684095	北京
Row1	personal	phone	1539684096	13111111111
Row11	personal	name	1539684094	李四
Row11	personal	city	1539684093	上海

Row Key	CF	CQ	Time Stamp	Value
Row1	office	tel	1539684043	010-1111111
Row1	office	address	1539684095	帝都大厦-18F-01
Row11	office	tel	1539684096	010-4444444
Row11	office	address	1539684094	帝都大厦-19F-02
Row2	office	tel	1539684093	010-3333333
Row2	office	address	1539684092	帝都大厦-18F-02

9 HBase架构

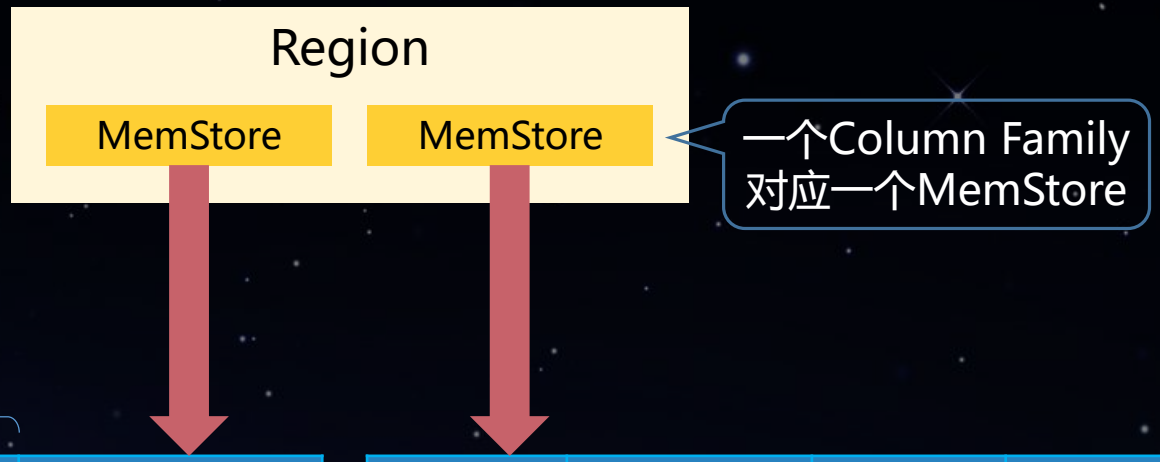

Client
Zookeeper
HMaster
HRegionServer
HRegion
HLog
Store
MemStore
StoreFile
HFile
HBase
DFS Client
HDFS
DataNode

10 HBase写数据



- 先将数据写到WAL中；
- WAL 存放在HDFS之上；
- 每次Put、Delete操作的数据均追加到WAL末端；
- 持久化到WAL之后，再写到MemStore中；
- 两者写完返回ACK到客户端。

11 HBase MemStore



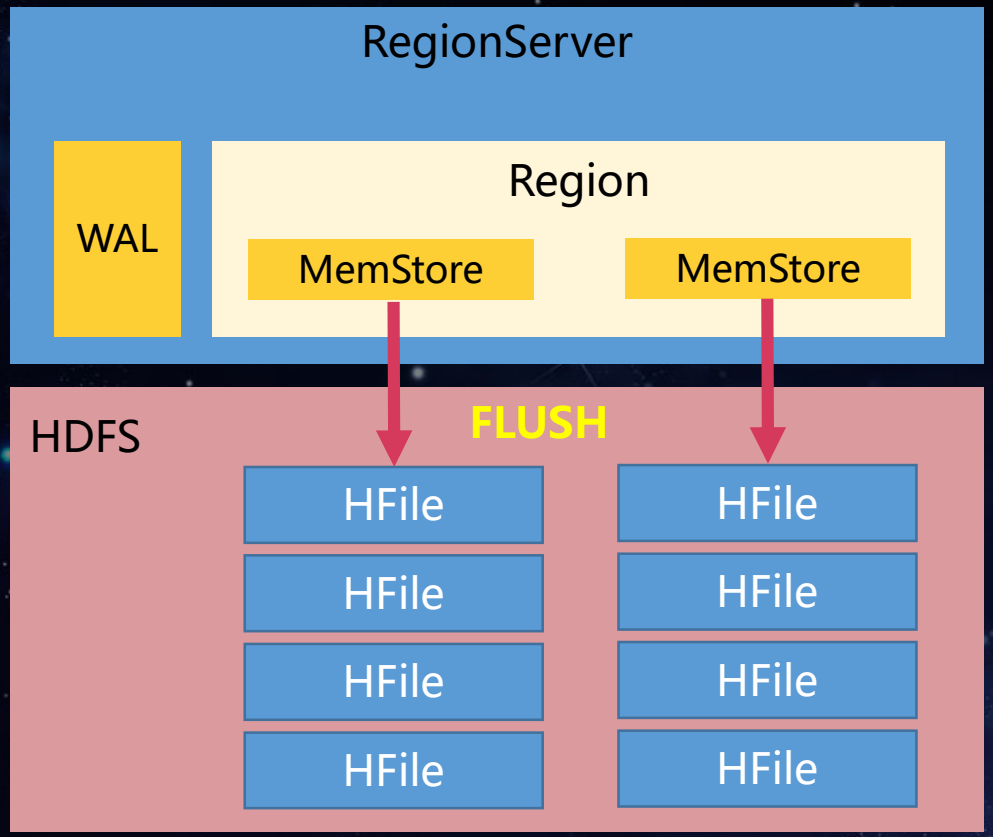
Key

Row Key	CF	CQ	Time Stamp	Value
Row1	personal	name	v1	张三
Row1	personal	city	v2	上海
Row1	personal	city	v1	北京
Row11	personal	name	v1	李四
Row11	personal	city	v1	上海
Row2	personal	name	v1	王五

Value

Row Key	CF	CQ	Time Stamp	Value
Row1	office	tel	v1	010-1111111
Row1	office	address	v1	帝都大厦-18F-01
Row11	office	tel	v1	010-4444444
Row11	office	address	v1	帝都大厦-19F-02
Row2	office	tel	v1	010-3333333
Row2	office	address	v1	帝都大厦-18F-02

12 HBase Region Flush



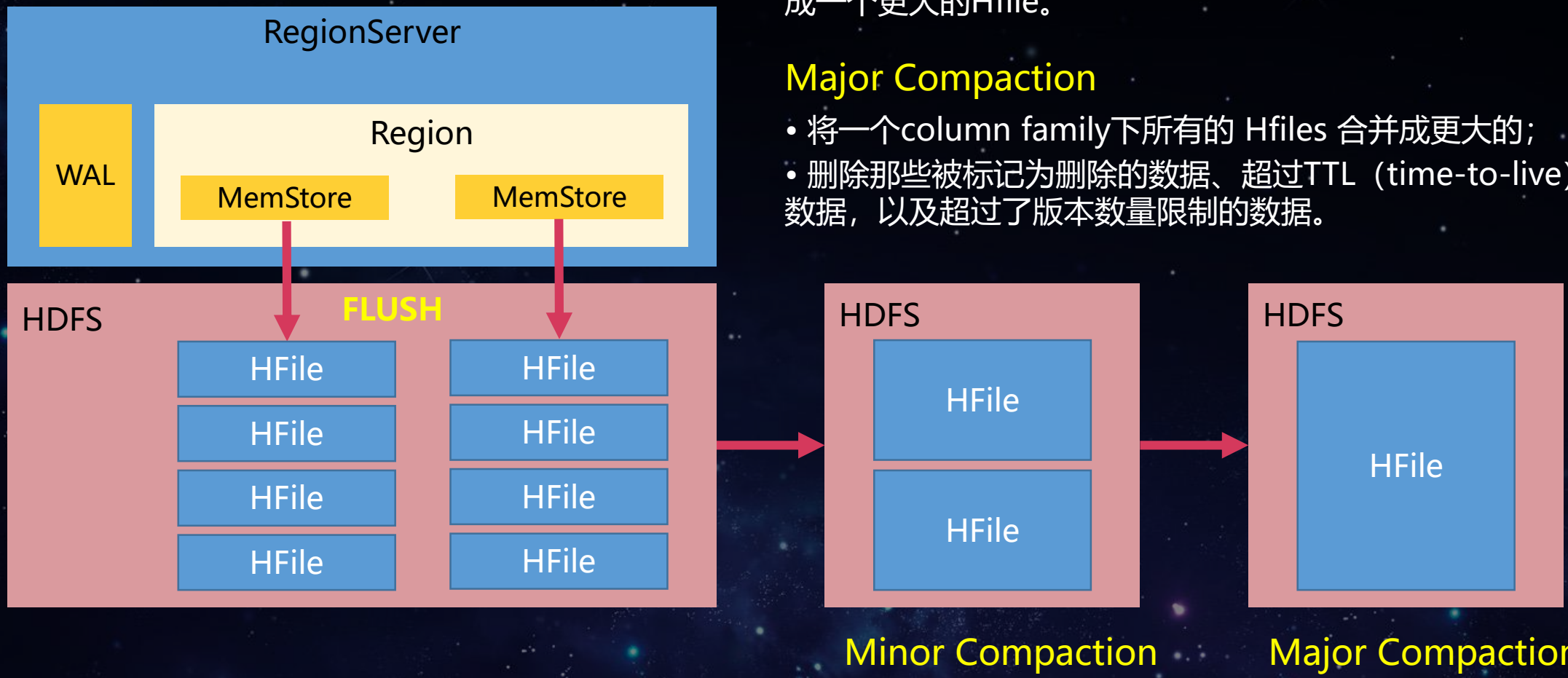
- 全局内存控制；
- MemStore使用达到上限；
- RegionServer的Hlog数量达到上限；
- 手动触发；
- 关闭RegionServer触发。

13 HBase Compaction

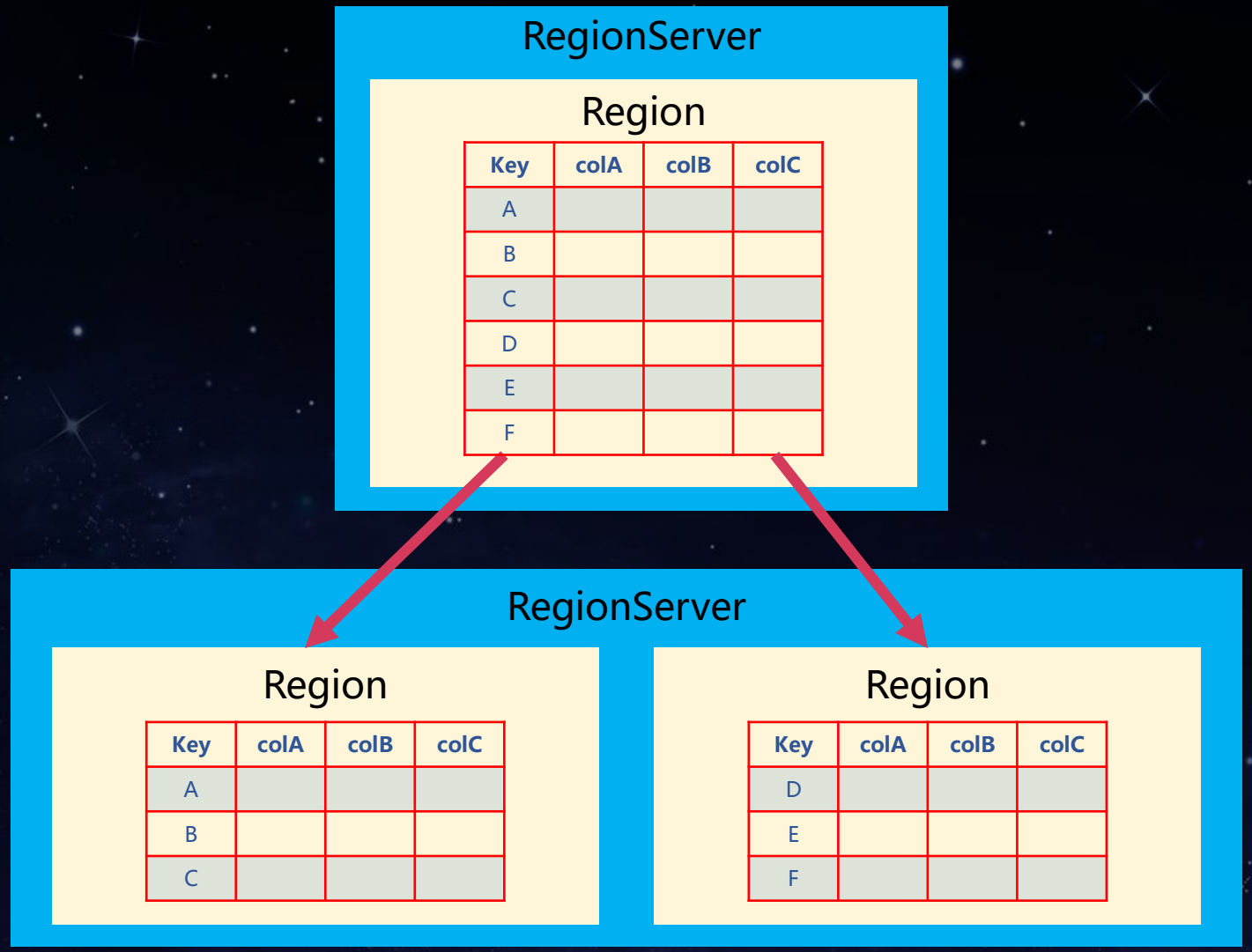
Minor Compaction: 指选取一些小的、相邻的HFile将他们合并成一个更大的Hfile。

Major Compaction

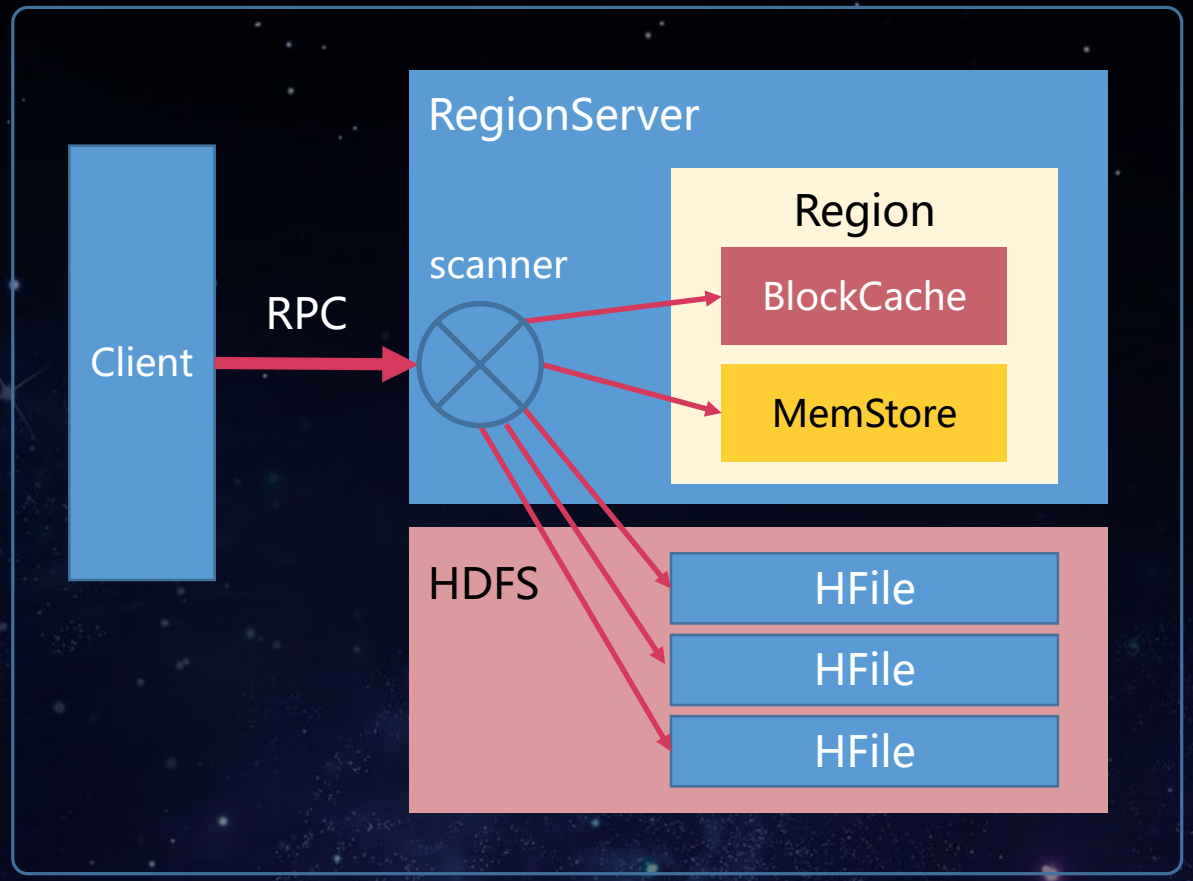
- 将一个column family下所有的 Hfiles 合并成更大的；
- 删除那些被标记为删除的数据、超过TTL（time-to-live）时限的数据，以及超过了版本数量限制的数据。



14 HBase Split



15 HBase读数据


RegionServer
Region
scanner
BlockCache
MemStore
Client
RPC
HDFS
HFile
HFile
HFile

16 HBase Region 查找

02 RowKey设计要点

1 RowKey的作用

- 读写数据时通过Row Key找到对应的Region
- MemStore 中的数据按RowKey字典顺序排序
- HFile中的数据按RowKey字典顺序排序

全局有序

RowKey	personal			office	
	name	city	phone	tel	address
Row1	张三	上海	13111111111	010-1111111	帝都大厦-18F-01
Row11	李四	上海		010-4444444	帝都大厦-19F-02
Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Row3	赵六		15166666666		帝都大厦-18F-03
Row4	孙七	北京		010-7777777	帝都大厦-18F-04
Row5	周八	深圳	15388888888		帝都大厦-18F-05
Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

2 RowKey的设计原则

RowKey	personal			office	
	name	city	phone	tel	address
Row1	张三	上海	13111111111	010-1111111	帝都大厦-18F-01
Row11	李四	上海		010-4444444	帝都大厦-19F-02
Row2	王五	武汉	18655555555	010-3333333	帝都大厦-18F-02
Row3	赵六		15166666666		帝都大厦-18F-03
Row4	孙七	北京		010-7777777	帝都大厦-18F-04
Row5	周八	深圳	15388888888		帝都大厦-18F-05
Row6	吴九	杭州		010-9999999	帝都大厦-18F-06
Row7	郑十	武汉	13599999999	010-5555555	帝都大厦-18F-07

全局有序

Region Region Region

结合业务的特点，并考虑高频查询，尽可能的将数据打散到整个集群。

3 RowKey的设计 - Salting

Salting 的原理是将固定长度的随机数放在行键的起始处

foo0001	afoo0001
foo0002	bfoo0002
foo0003	cfoo0003
foo0004	dfoo0004

优缺点：由于前缀是随机生成的，因而如果想要按照字典顺序找到这些行，则需要做更多的工作。从这个角度上看，salting增加了写操作的吞吐量，却也增大了读操作的开销。

4 RowKey的设计 - Hashing

Hashing 的原理将RowKey进行hash计算，然后取hash的部分字符串和原来的RowKey进行拼接。

foo0001	aaafoo0001
foo0002	bbbfoo0002
foo0003	cccfoo0003
foo0004	dddfoo0004

优缺点：可以一定程度打散整个数据集，但是不利于Scan；由于不同数据的hash值可能一样，实际应用中可以使用md5计算，然后截取前几位的字符串。如下：

subString(MD5(设备ID), 0, x) + 设备ID，其中x一般取5或6。

5 RowKey的设计 - Reversing

Reversing 的原理是反转一段固定长度或者全部的键

abc.iteblog.com	moc.golbeti.cba
www.iteblog.com	moc.golbeti.www
cdn.iteblog.com	moc.golbeti.ndc
def.iteblog.com	moc.golbeti.fed

优缺点：有效地打乱了行键，但是却牺牲了行排序的属性。

03 HBase多模式

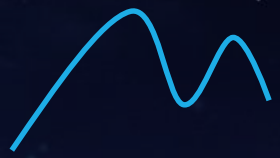
1 多种数据格式

Key - Value

关系数据

文档数据

图数据

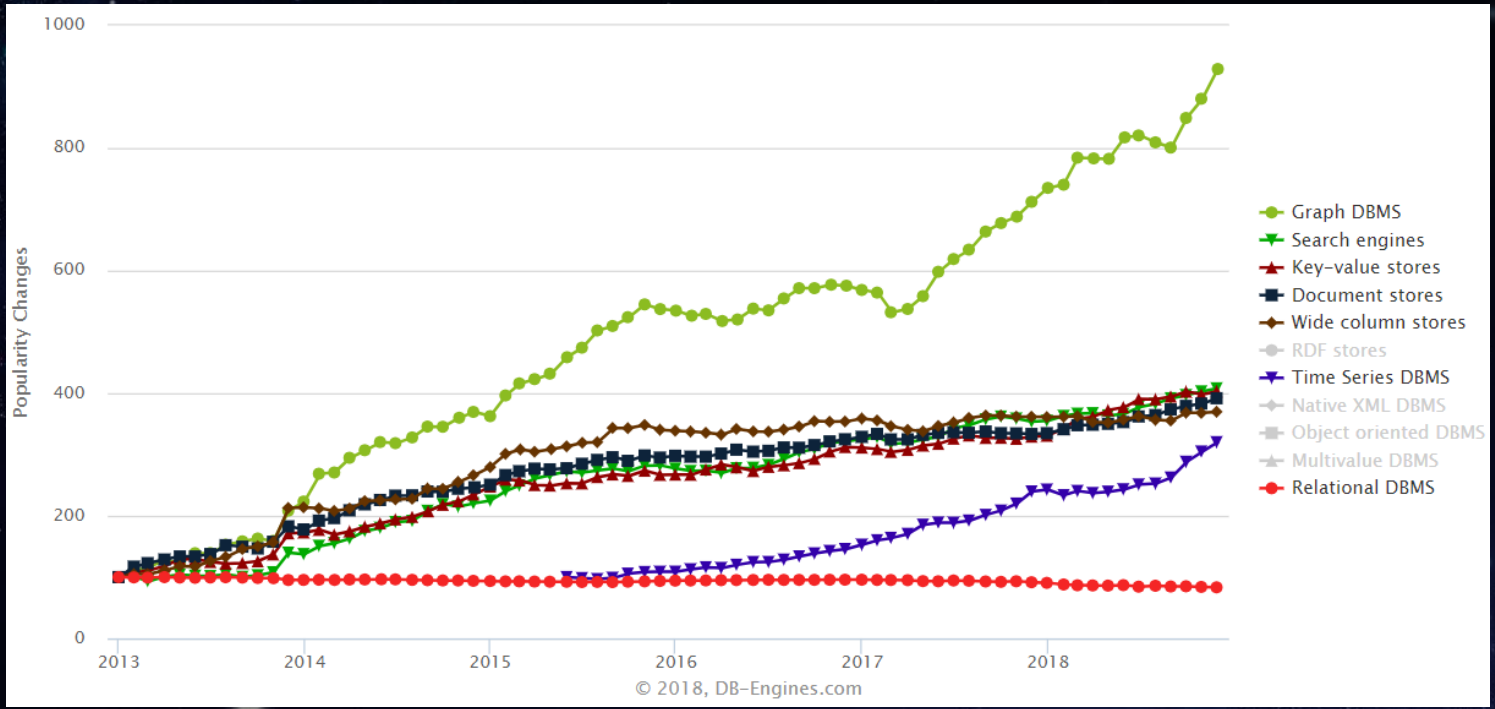


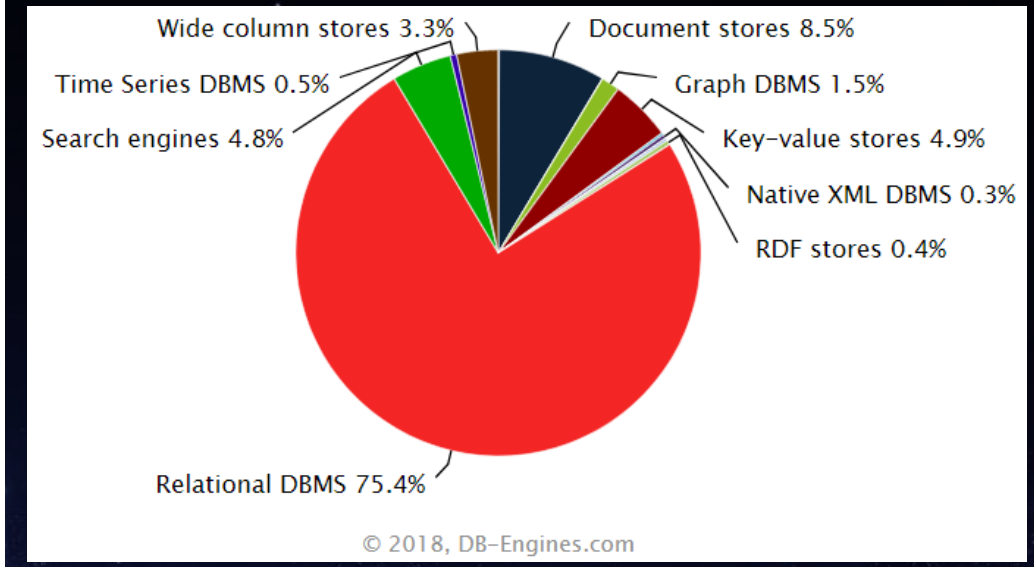
时序数据

时空数据

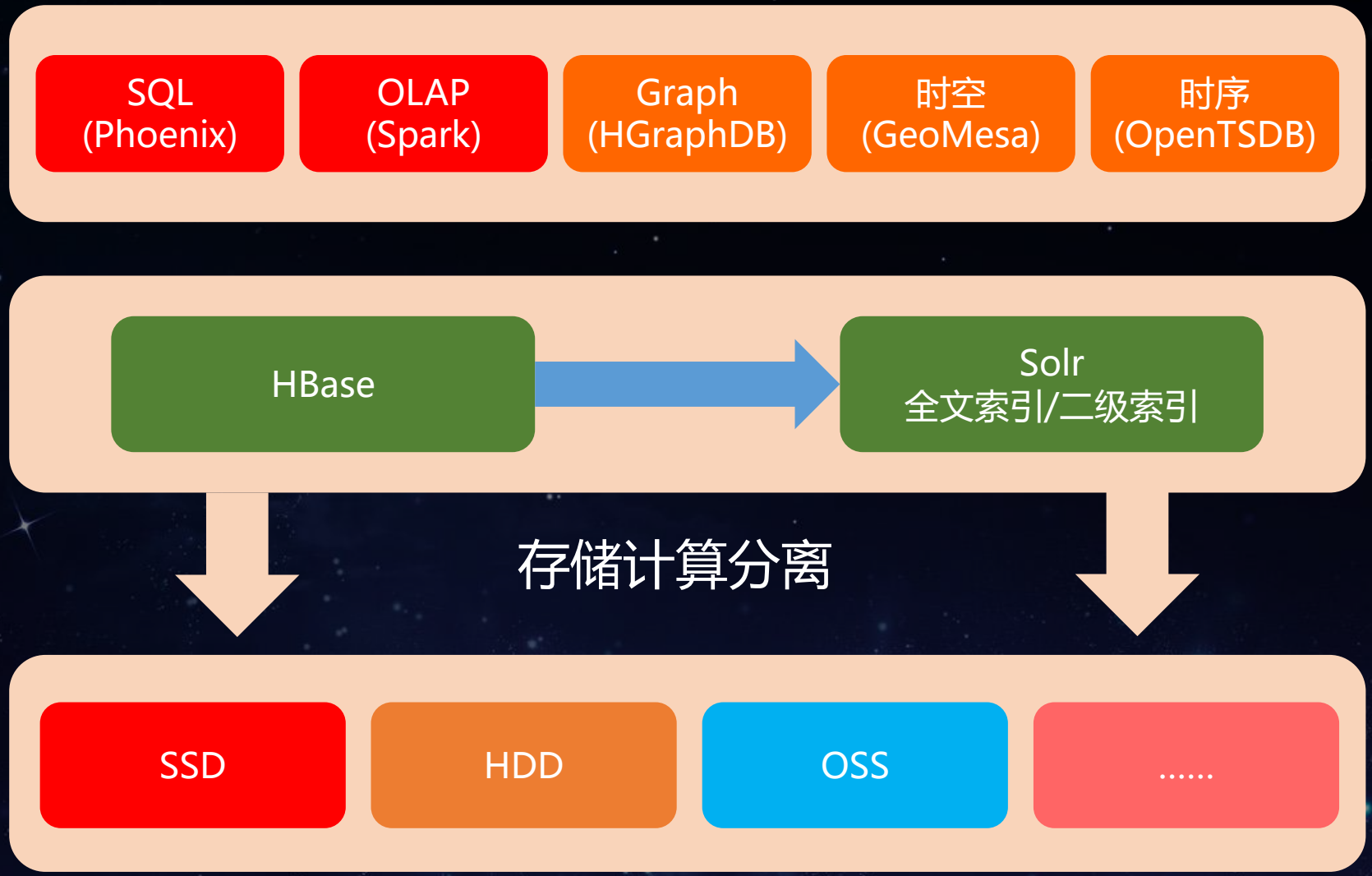
.....

2 多种数据格式


1000
800
600
400
200
0
Popularity Changes
2013
2014
2015
2016
2017
2018
Graph DBMS
Search engines
Key-value stores
Document stores
Wide column stores
RDF stores
Time Series DBMS
Native XML DBMS
Object oriented DBMS
Multivalue DBMS
Relational DBMS
© 2018, DB-Engines.com


Wide column stores 3.3%
Document stores 8.5%
Time Series DBMS 0.5%
Graph DBMS 1.5%
Search engines 4.8%
Key-value stores 4.9%
Native XML DBMS 0.3%
RDF stores 0.4%
Relational DBMS 75.4%
© 2018, DB-Engines.com

3 HBase多模式


SQL
(Phoenix)
OLAP
(Spark)
Graph
(HGraphDB)
时空
(GeoMesa)
时序
(OpenTSDB)
HBase
Solr
全文索引/二级索引
存储计算分离
SSD
HDD
OSS
.....

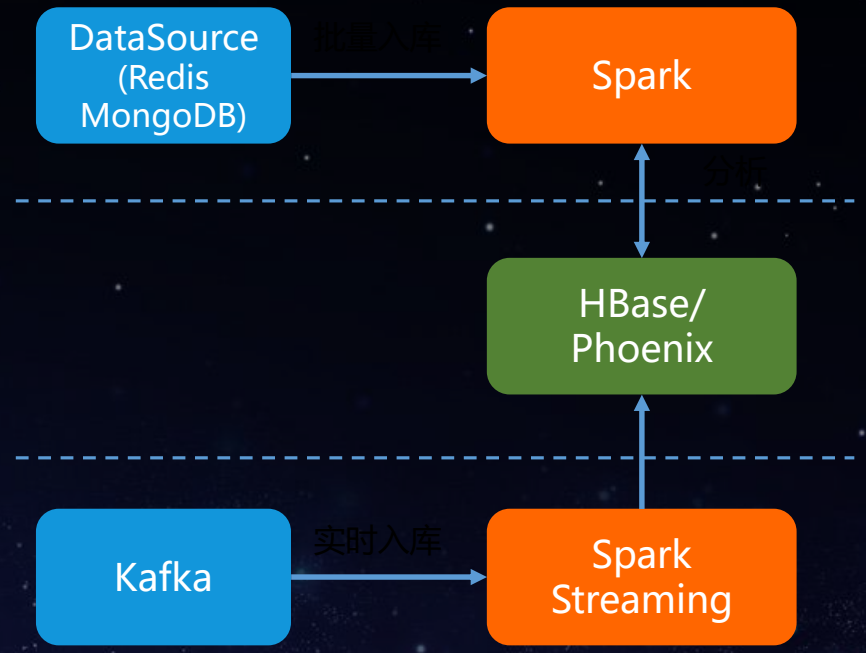
4 Phoenix - SQL

- 为HBase提供关系型数据库的SQL能力，用于低延迟的OLTP及操作性分析
- SQL引擎
  - 支持ANSI 92语法
  - 算子/过滤条件下推到Server执行
  - 支持所有HBase特性：动态列、TTL等
  - 支持二级索引

	MySQL	Phoenix
数据库类型	RDBMS	RDBMS & KV
WorkLoad	OLTP	TP&操作性AP
扩展能力	单机	横向扩展
数据量	TB	PB
二级索引	支持	支持
事务	ACID	行级别事务

5 Spark – 分析

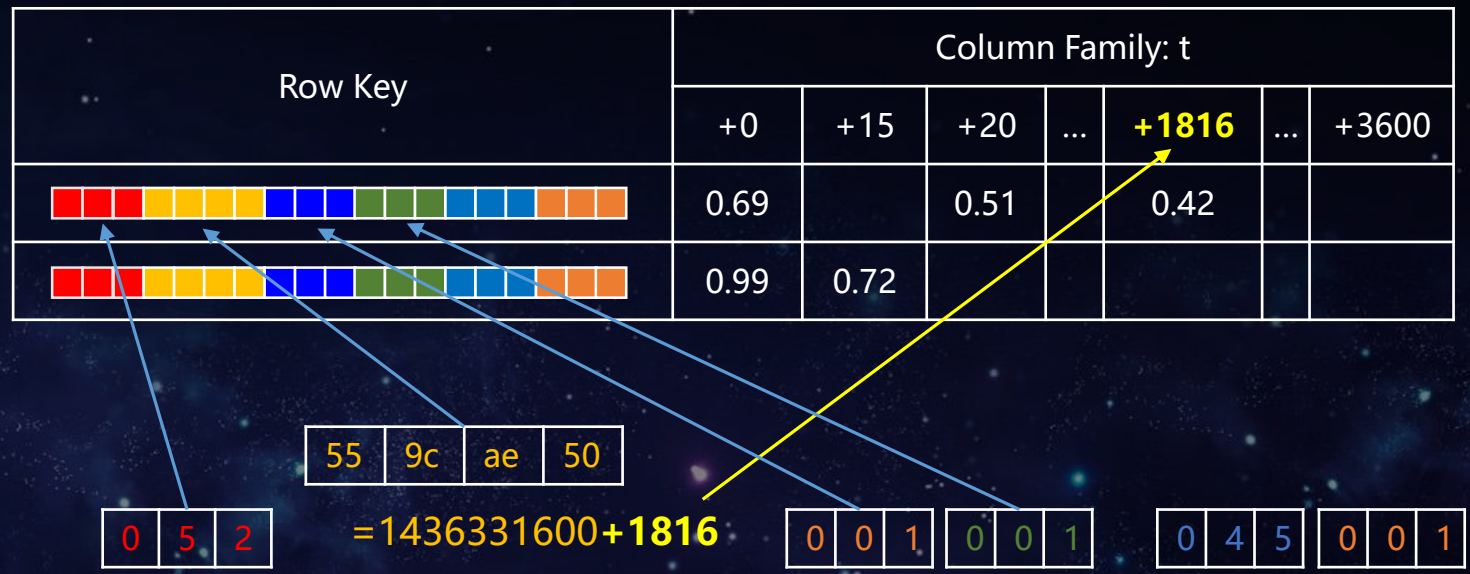
- OLAP；
- 利用 Spark-SQL 查询一些比较复杂的分析；
- 利用 Spark Streaming 进行实时流分析，结果存入HBase；
- 直接读取HFile。



5 OpenTSDB - TimeSeries

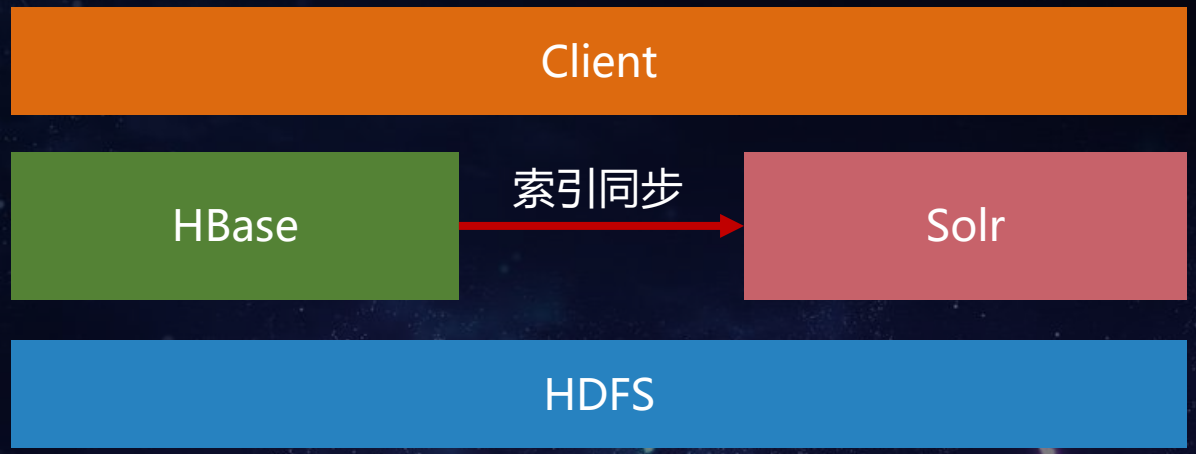
- 基于HBase的分布式的，可伸缩的时间序列数据库；
- 主要满足物联网：传感器、监控、金融K线等时序应用场景
- 海量数据存储、高并发高吞吐写；
- 压缩节约存储空间 - 包括多行变一行及多列压为一列
- Salt 机制保障没有热点

```
proc.loadavg.1m 1436333416 0.42 host=web42 pool=static
proc.loadavg.5m 1436333416 0.26 host=web42 pool=static
proc.loadavg.1m 1436333417 0.49 host=web42 pool=static
proc.loadavg.5m 1436333417 0.30 host=web42 pool=static
proc.loadavg.1m 1436333418 0.39 host=web42 pool=static
proc.loadavg.5m 1436333418 0.23 host=web42 pool=static
```



6 Solr

- 基于Lucene的全文搜索引擎；
- 为HBase添加二级索引功能；
- 提供范围查找、模糊查找等。



7 HGraphDB - Graph

- HGraphDB是分布式图数据库，底层基于HBase；
- 支持数百亿点与边的即时查询；
- 支持OLTP & OLAP。

应用：依托图关联技术，帮助金融机构有效识别隐藏在网络中的黑色信息，在团伙欺诈、黑中介识别等。

8 GeoMesa - GeoSpatial

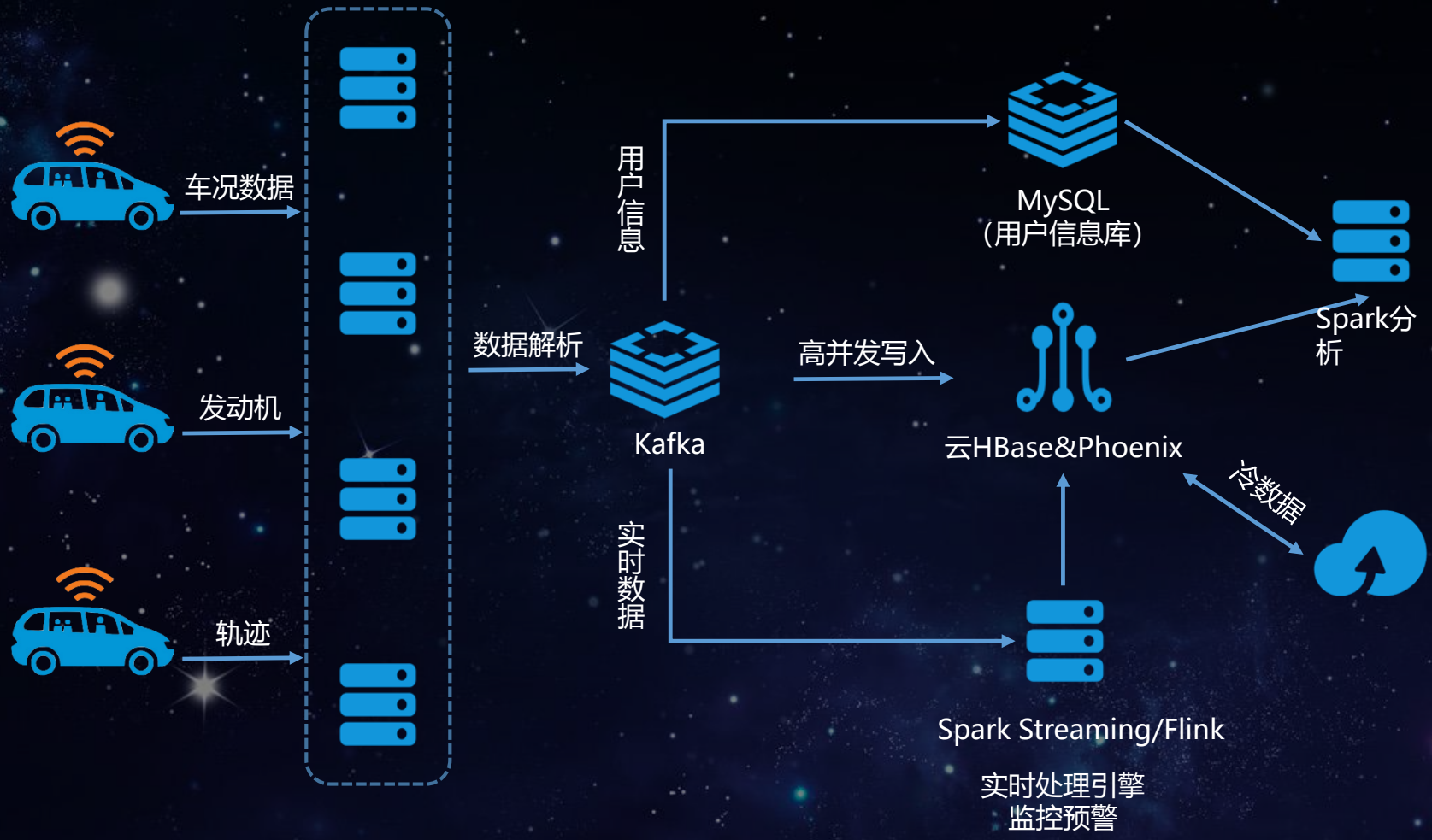
- 目前基于NoSQL数据库的时空数据引擎中功能最丰富、社区贡献人数最多的开源系统；
- 提供了多种空间索引方式供用户灵活选择；
- 提供了基于Coprocessor的空间查询方式，将计算过程放置在server端，能够减少通讯开销，从而获得很好的性能提升；
- 提供了丰富数据入库、操作等工具，便于用户处理数据；
- 提供了多种空间数据分析算法，如KNN、直方图、热点分析、TubeSelect等；
- 基于OGC标准设计，便于系统间的集成与互操作。

04 HBase典型案例分析

1 HBase应用场景

2 车联网案例 – 海量数据存储

场景：百万车载终端，百TB级数据不间断写入，数十亿级数据量下分页查询和车辆历史轨迹查询要求毫秒级响应



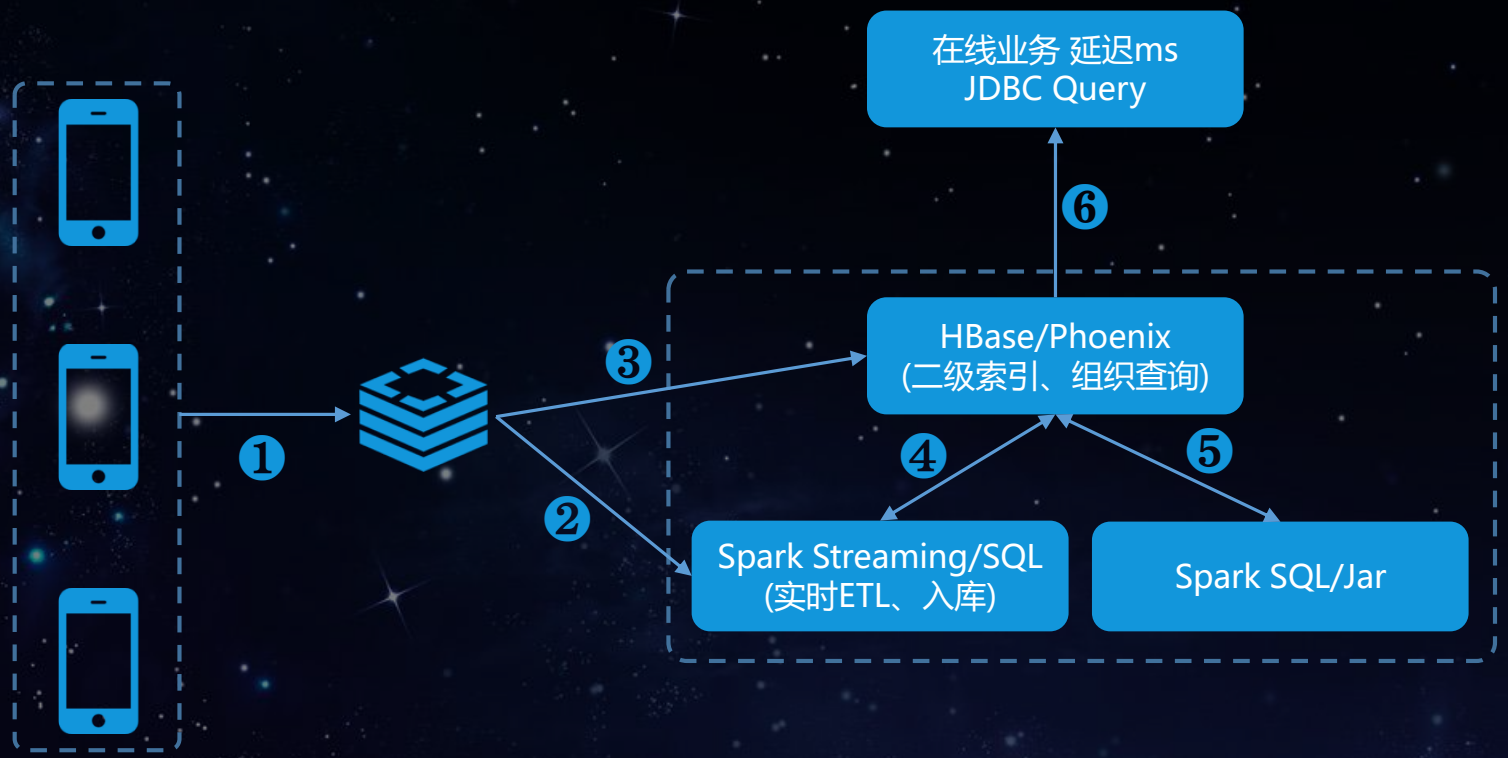
痛点：日均百GB级别，全量数据TB级别，冷数访问频率低。

解决方案：支持冷数据存放在OSS之中，使用ZSTD压缩算法，降低3倍存储成本。

痛点：车载终端数据高并发写入，传统数据库延迟高

解决方案：HBase采用LSM存储模型，适合物联网高并发写入，同时满足高并发读

3 广告买流分析平台



- 准实时分析广告投放转化率，不断优化广告投放策略
- 准实时分析用户行为并更新用户画像数据，不断优化广告推荐算法
- 实时处理海量用户手游端上传的数据（类似物联网）
- 全托管HBase及Spark服务，免维护，不需要频繁监控所有节点的健康情况和调度，由平台统一负责监控、管理和调度

一、数据源

(1)：捕获用户行为事件或者手游端上传的数据并写入Kafka集群；

二、实时入库及ETL

(3)：从Kafka读取手游端上传的数据然后直接写入HBase/Phoenix；

(2)、(4)：Spark streaming实时读取Kafka数据，在做ETL的同时读取HBase/Phoenix的维表数据；并将结果数据写入HBase/Phoenix对外提供查询；

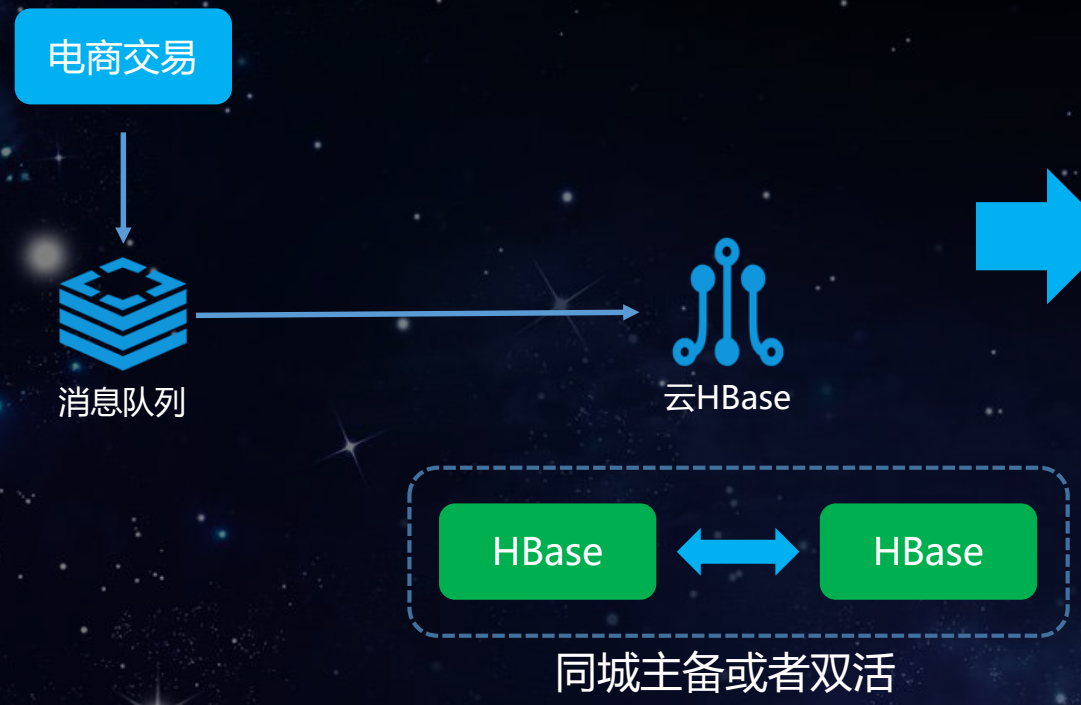
三、复杂分析

(5)：Spark SQL对HBase/Phoenix中的数据做复杂分析，并把结果写入HBase/Phoenix；

四、在线查询

(6)：BI或个性化推荐系统通过标准JDBC接口，实时查询HBase/Phoenix中的用户画像数据。

4 支付宝账单查询



痛点1：查询要求 999 延迟 50ms，毛刺尽量少，平均延迟2ms以内

解决方案：通过基于OffHeap及AliGC技术，让YGC从120ms降低为15ms左右，大幅度降低毛刺

痛点2：历史订单数据量较多，存储成本较贵

解决方案：云HBase基于共享存储，让全局副本数从基于云盘的9副本降低为3副本，成本降低60%+；HBase本身最高10倍压缩比例降低成本

痛点3：在线查询需要预防灾难发生，如果机房故障可能业务中断

解决方案：云HBase提供同城主备或者双活的方式，保障机房级别容灾

5 历史订单全文查找

某订单信息表：378列，其中13个列，需要模糊查询

	索引查询条件说明	查询示例	平均查询时间(ms)	Solr查询时间
	精确查询	qd_s:QD030000545472	91	34
	模糊查询	sj_s:SJ0600004*613	303	228
	范围查询	rq_i:[100000 TO 200000]	252	180
AND组合	精确查询	rq_i:2571198 AND yd_s:YD040000657960	86	73
	模糊查询	xm_s:EEE?MP*FFFFF AND sj_s:SJ0800007*123	358	342
	范围查询	rq_i:[8000 TO 82000] AND zt_s:ZT0000000004	343	259
	组合查询	ys_s:YS0000000004 AND xm_s:LLL*Q*MMMMM AND rq_i:[80000 TO 9000000]	611	530
OR组合	精确查询	rq_i:175948 OR rq_i:175971	223	160
	模糊查询	xm_s:AAA?MP*FFFFF OR xm_s:AAA*AE*FFFFF	249	239
	范围查询	rq_i:[175900 TO 175950] OR rq_i:[175951 TO 17600]	197	172
	组合查询	rq_i:175948 OR xm_s:AAA?MP*FFFFF OR xm_s:AAA*AE*FFFFF OR rq_i:[175900 TO 175950]	220	212
AND+OR组合	精确查询	(rq_i:175948 OR rq_i:175971) AND yd_s:YD100000583175	76	62
	模糊查询	(xm_s:GGG?MP*MMMMM AND xm_s:GGG*E*MMMMM) OR xm_s:GGG?QG*MMMMM	233	226
	范围查询	(rq_i:[8000 TO 82000] AND rq_i:[9000 TO 10000]) OR rq_i:[175951 TO 17600]	206	198
	组合查询	(xm_s:JJJ*BQ*CCCCC OR rq_i:[1320000 TO 1400000]) AND xm_s:JJJ*AE*CCCCC	259	252

架构要点：

- 全量数据与索引数据分开，大概30:1数据量关系。比如原始数据30T，索引数据1T左右；
- Solr数据尽量缓存在内存；
- 云HBase内置同步逻辑，保障数据不丢失；
- 90%查询走HBase API，10%走solr。

CHTC
中国HBase技术社区

云栖社区

THANK YOU

中国 HBase 技术社区网站：http://hbase.group

CHTC
中国HBase技术社区

云栖社区

社区管理员

HBase 技术社区公众号

中国 HBase 技术社区网站：http://hbase.group