



海量数据实时处理框架HBASE

主要内容

- HBase架构原理
- HBase高级应用
- HBase安装部署
- HBase开发设计
- HBase运营维护
- HBase参数配置和调优
- HBase案例分析

HBase架构原理-基本概念

HBase简介与架构

- HBase是一个分布式的、面向列的开源数据库，该技术来源于Chang et al所撰写的Google论文"Bigtable：一个结构化数据的分布式存储系统"
- 高可靠性、高性能、面向列、可伸缩
- HBase-Hadoop Database
- HDFS为HBase提供高可靠底层存储支持
- MapReduce为HBase提供高性能计算能力
- Zookeeper为HBase提供稳定服务和failover机制

HBase简介与架构

- HBase是Google Bigtable的开源实现，两种实现对比：
 - HDFS - GFS
 - Hadoop MapReduce - MapReduce
 - Zookeeper - Chubby

HBase简介与架构

版本沿革

- 2006-11 Google releases paper on BigTable
- 2007-02 HBase作为Hadoop contrib原型
- 2008-01 第一个不稳定版本 HBase (Hadoop 0.15.0)
- ...

HBase简介与架构

版本沿革

- hbase-0.1.0 2008-03-27 release
- hbase-0.2.0 2008-08-08 release
- ...
- hbase-0.18.0 2008-9-21 release
- hbase-0.20.6 2010-07-10 release
- hbase-0.89.20100621 2010-06-25 release
- ...

HBase简介与架构

- hbase-0.90.0
- ...
- hbase-0.92.0
- ...
- hbase-0.94.0
- ...

HBase简介与架构

- hbase-0.94.6.1 old stable
- hbase-0.94.7 old stable
- hbase-0.94.18 stable version
- hbase-0.96 current develop
- 开发者是Jim Kellerman、Michael Stack和Bryan Duxbury

HBase简介与架构

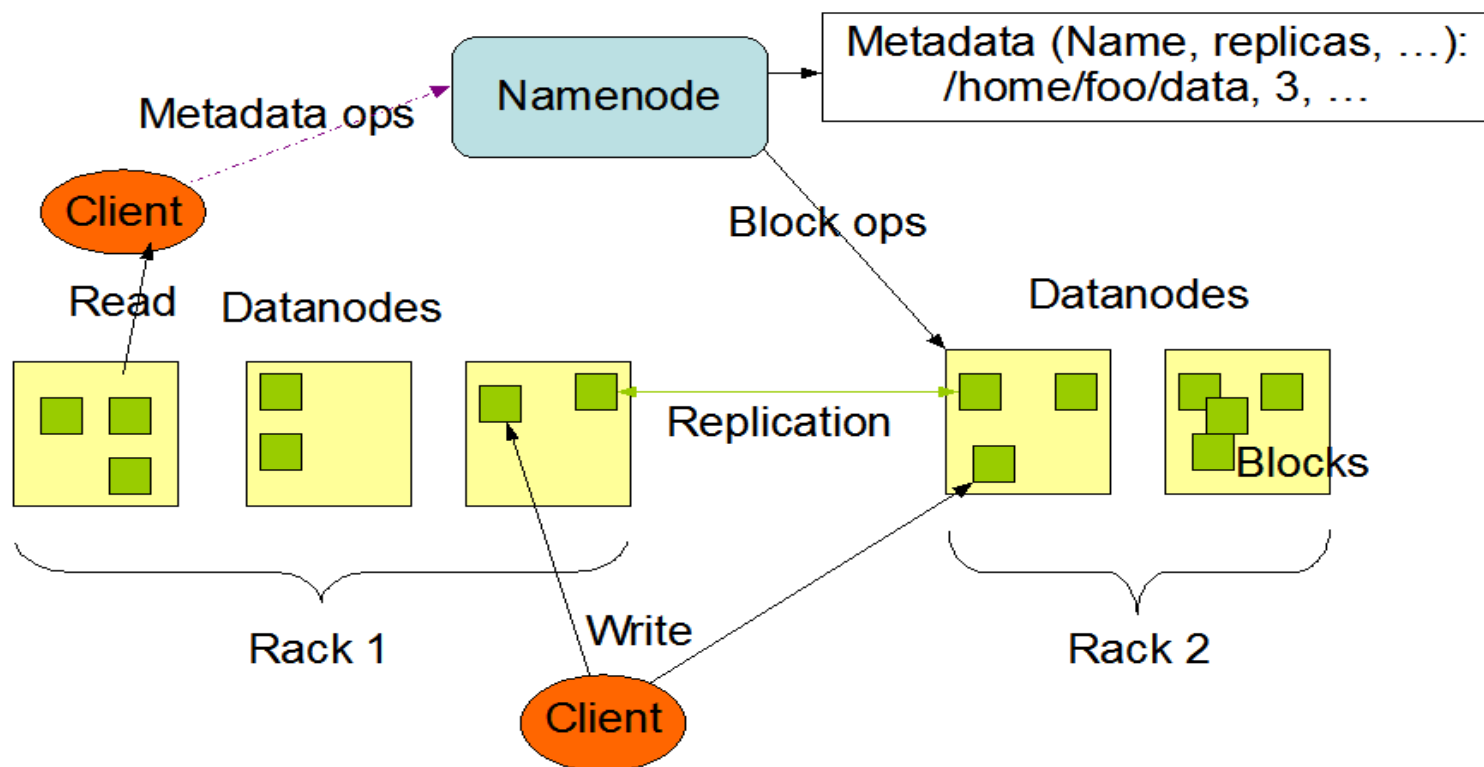
- hbase-0.20.6 -> hbase-0.89
- 很大的版本跳跃，原因：
 - 不在跟hadoop步伐一致
 - 更接近1.0，基本实现bigtable功能
- 更多参见：

<http://wiki.apache.org/hadoop/Hbase/HBaseVersions>

HBase简介与架构

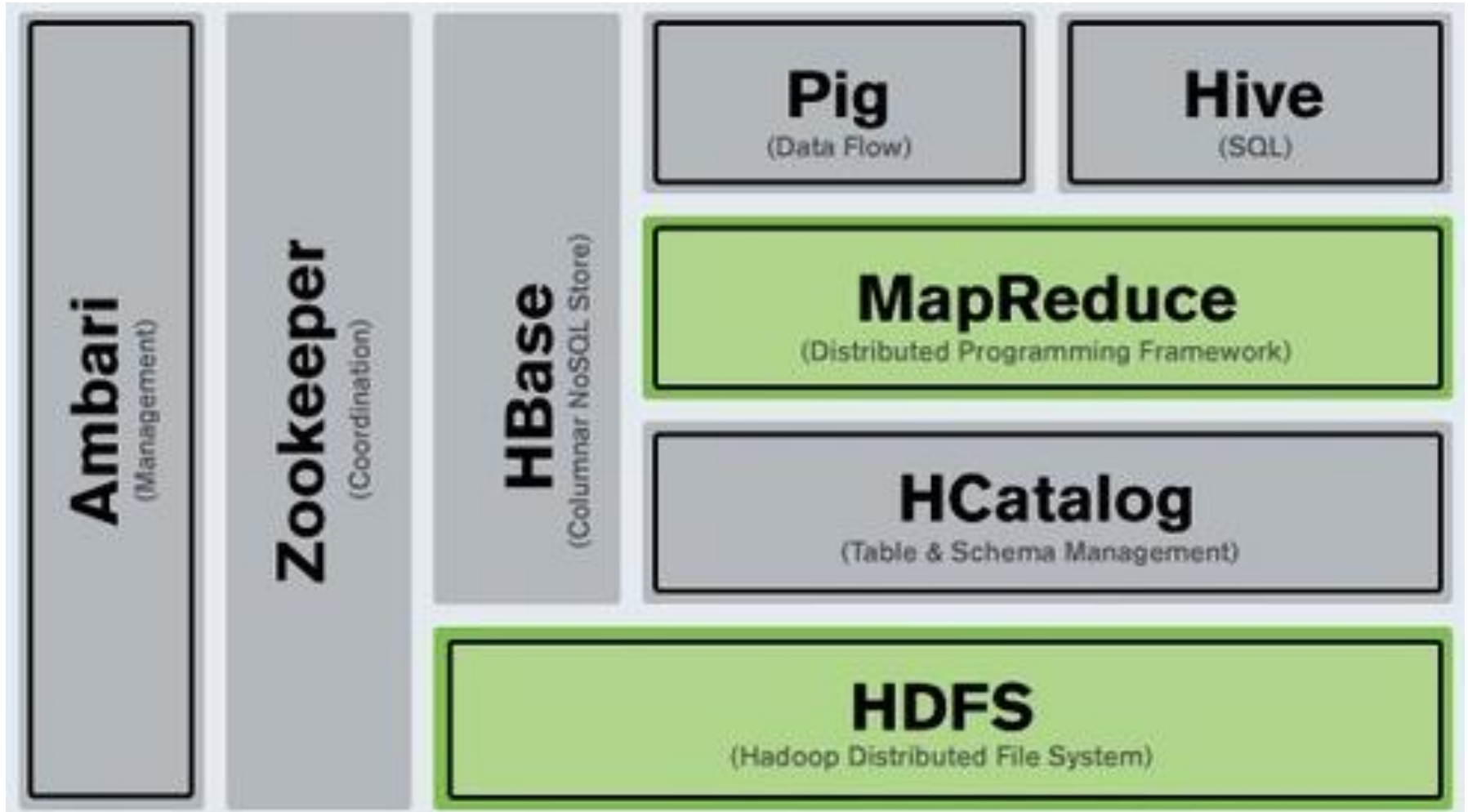
- HDFS为HBase提供高可靠底层存储支持

HDFS Architecture



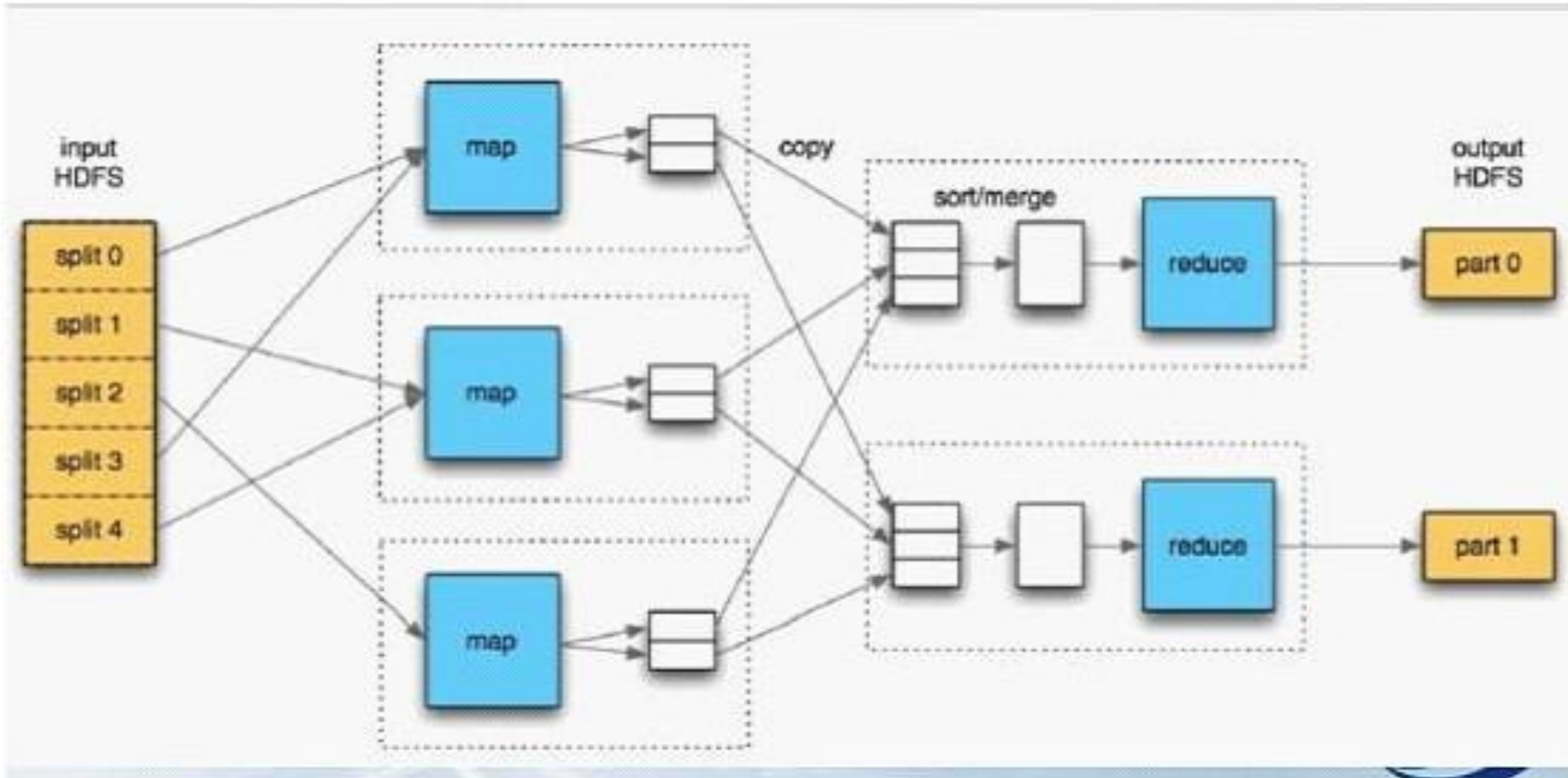
HBase简介与架构

- MapReduce提供高性能计算能力



HBase简介与架构

- MapReduce提供高性能计算能力



HBase简介与架构

- Zookeeper 消息协调通信 --Yahoo!
- ZooKeeper是一个高可用的分布式数据管理与系统协调框架。基于对Paxos算法的实现，使该框架保证了分布式环境中数据的强一致性

HBase简介与架构

- 数据发布与订阅
- Name Service
- 分布通知/协调
- 分布式锁
- 集群管理
- 分布式队列

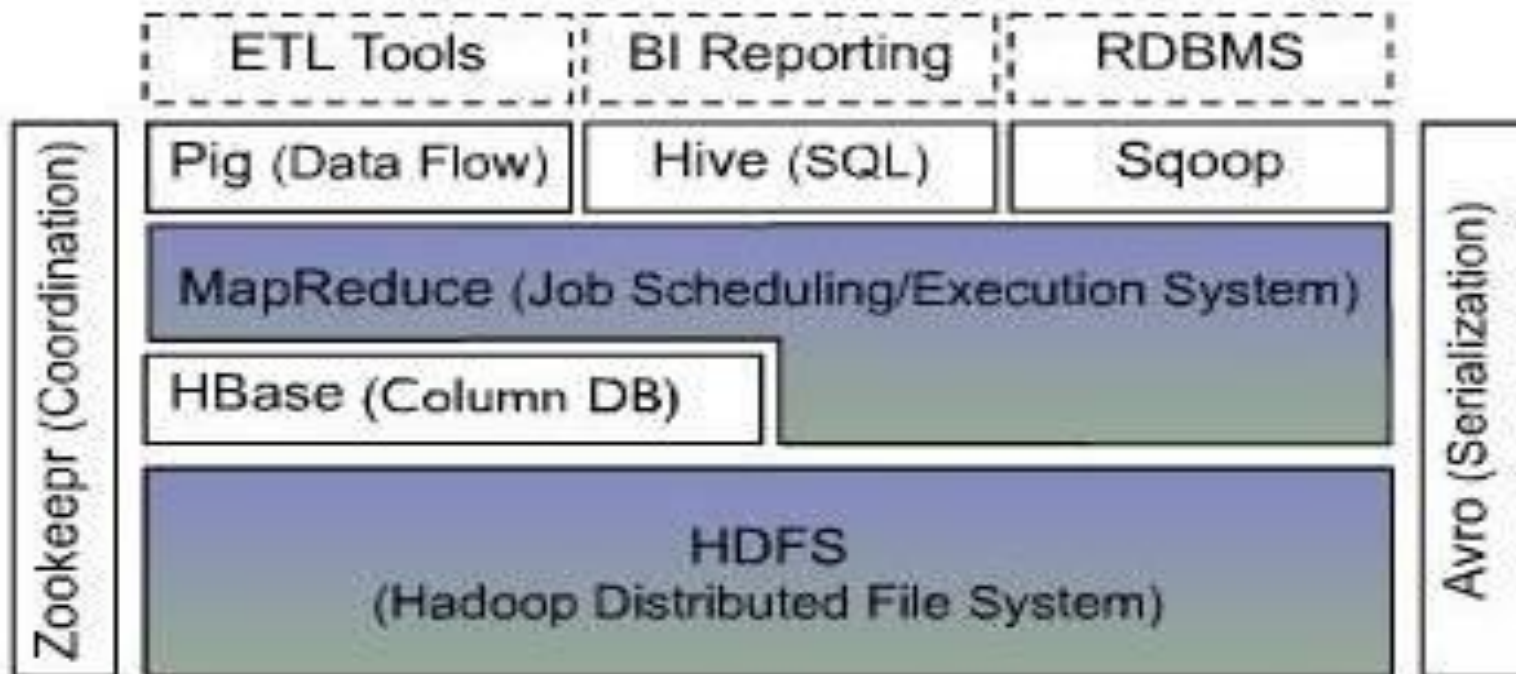
HBase简介与架构

NoSQL

- NoSQL - Not Only SQL
- 海量数据
- 保证一定的查询效率
- 弱关系型、松散结构
- 分布式存储

HBase简介与架构

The Hadoop Ecosystem



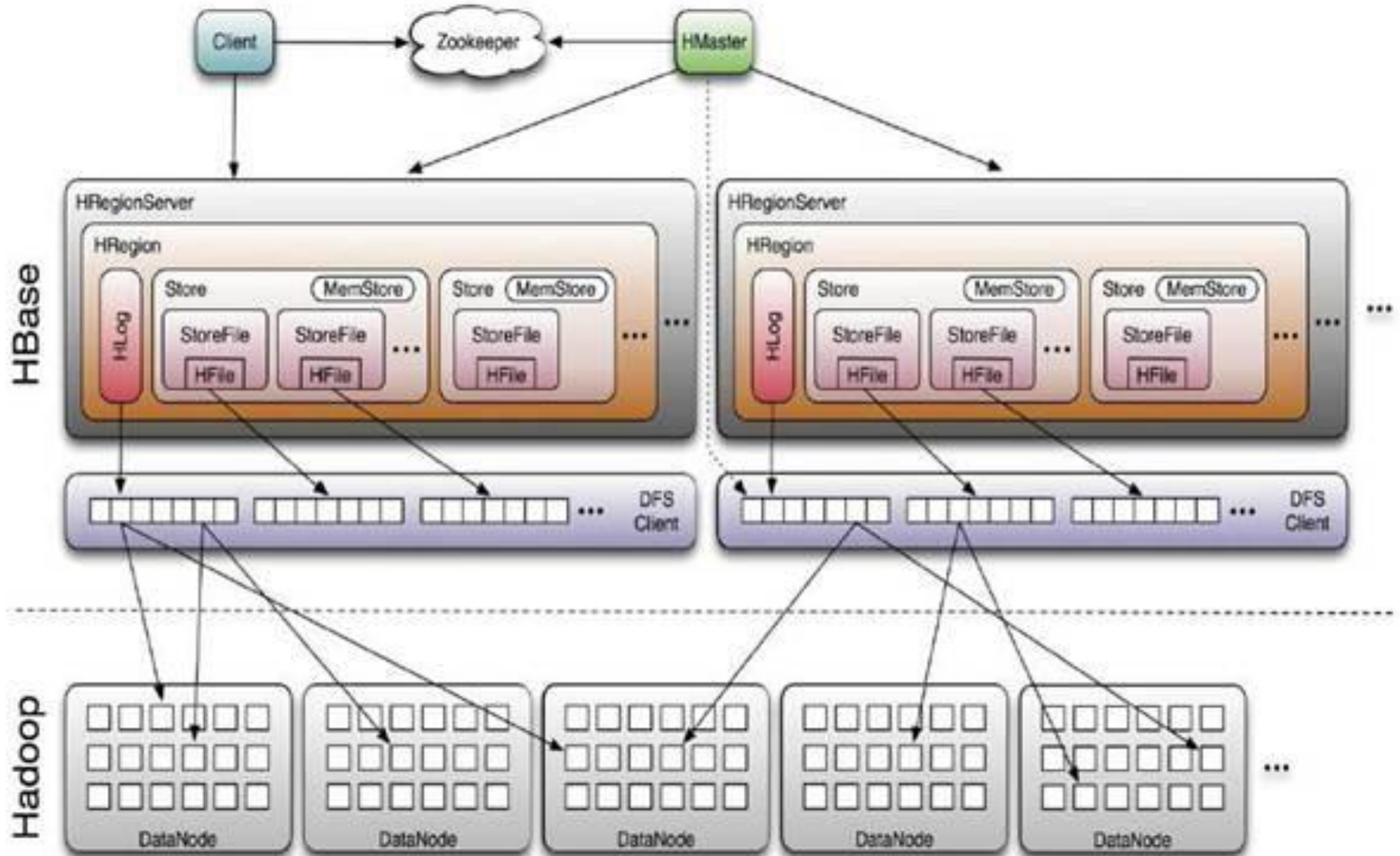
HBase使用场景

- 存储大量的数据（ >TB ）
- 需要很高的写吞吐量
- 大规模数据集很好性能的随机访问（ 按列 ）
- 需要进行优雅的数据扩展
- 结构化和半结构化的数据
- 不需要全部的关系数据库特性，例如交叉列、交叉表，事务，连接等等

HBase简介与架构

- Client – RPC(management, CRUD)
- Zookeeper – Quorum,协调通信
- HMaster – table/region的管理
- HRegionServer – 核心,响应用户I/O,向HDFS读写数据

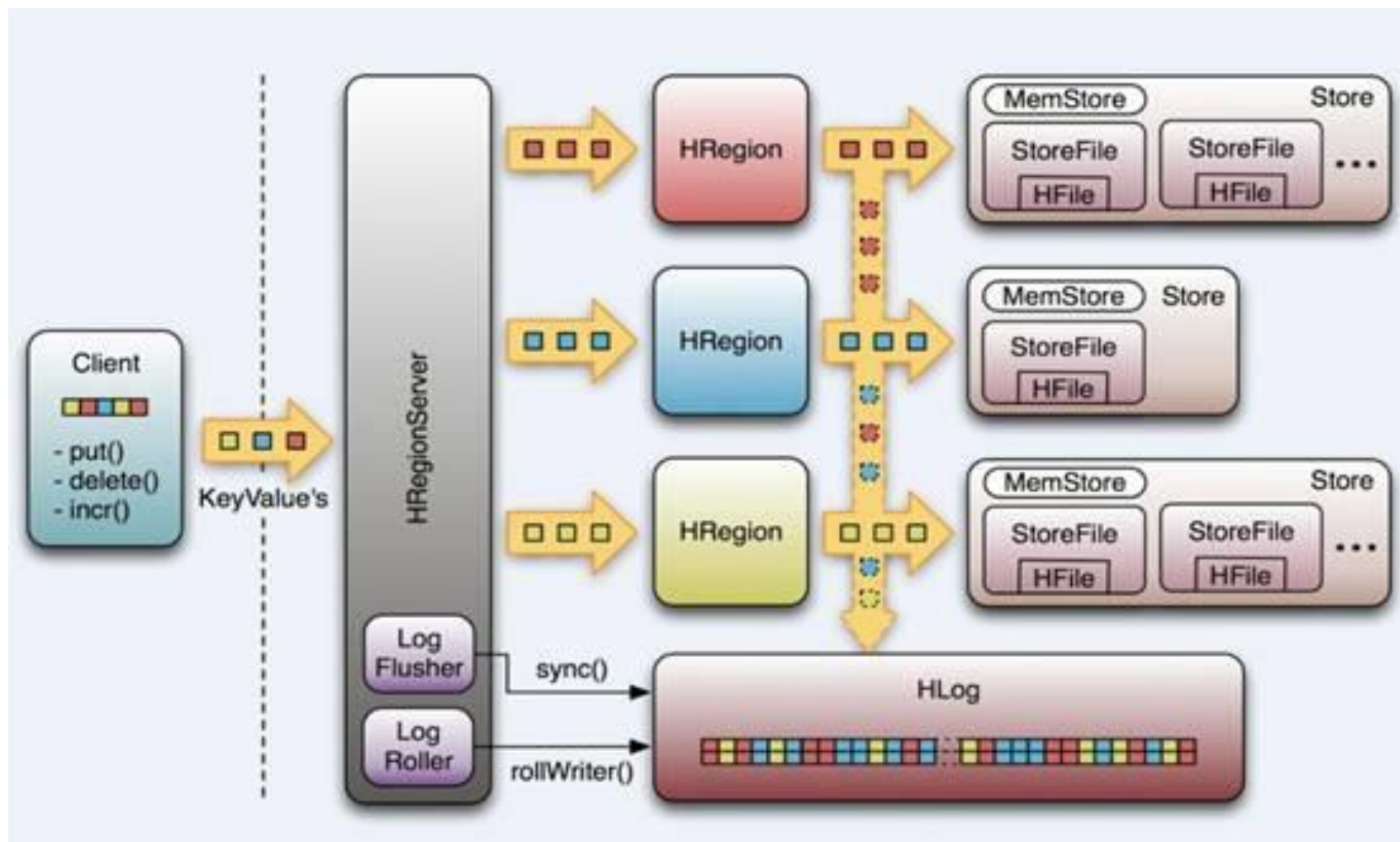
HBase简介与架构



HBase简介与架构-HMaster

- 管理用户对Table的增、删、改、查操作
- 管理HRegionServer的负载均衡，调整Region分布
- 在Region Split后，负责新Region的分配
- 在HRegionServer停机后，负责失效HRegionServer 上的Regions迁移

HBase简介与架构-HRegionServer



HBase简介与架构-HRegionServer

- HRegionServer内部管理了一系列HRegion对象
- 每个HRegion对应了Table中的一个Region
- HRegion中由多个HStore组成
- 每个HStore对应了Table中的一个Column Family的存储

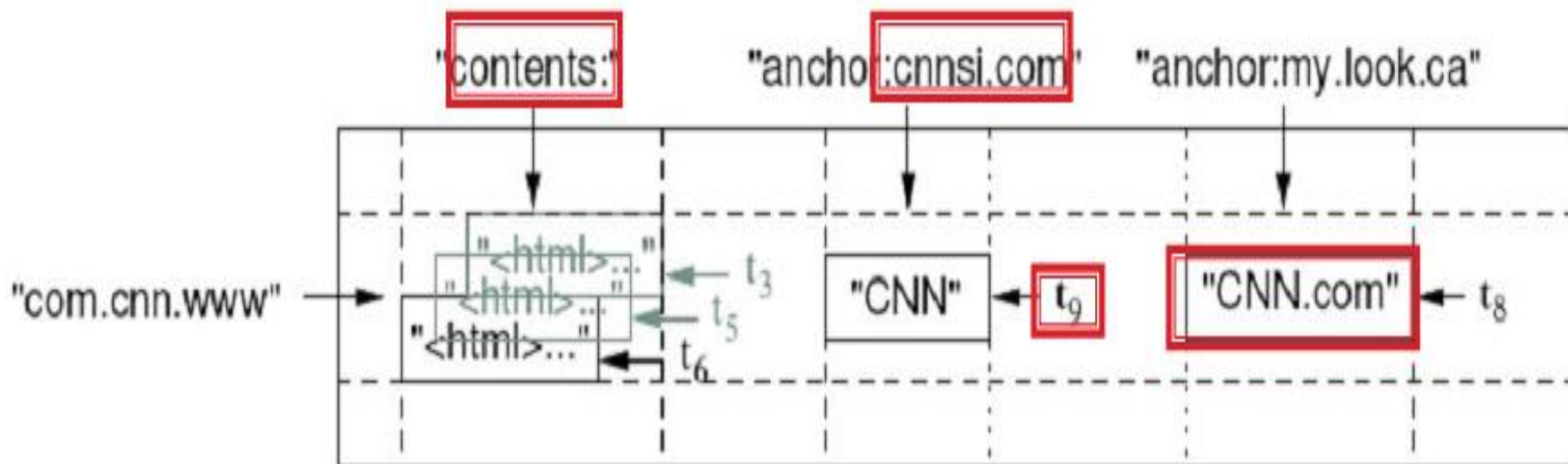
HBase简介与架构-HRegionServer

- HStore存储是HBase存储核心，其中由两部分组成，一部分是
- MemStore: Sorted Memory Buffer，用户写入的数据首先会放入MemStore
- StoreFiles: 当MemStore满了以后会Flush成一个StoreFile（底层实现是HFile）

HBase简介与架构-HRegionServer

- 当StoreFile文件数量增长到一定阈值，会触发Compact合并操作，将多个StoreFiles合并成一个StoreFile，合并过程中会进行版本合并和数据删除
- 因此可以看出HBase其实只有增加数据，所有的更新和删除操作都是在后续的compact过程中进行的

HBase简介与架构-数据逻辑结构



HBase简介与架构-数据逻辑结构

- Rowkey - 行键（主键）
- Column family - 列簇
- Timestamp - 时间戳（版本）
- Cell - {row key, column(= <family> + <qualifier>), version}唯一确定，无数据类型，全部是字节码形式

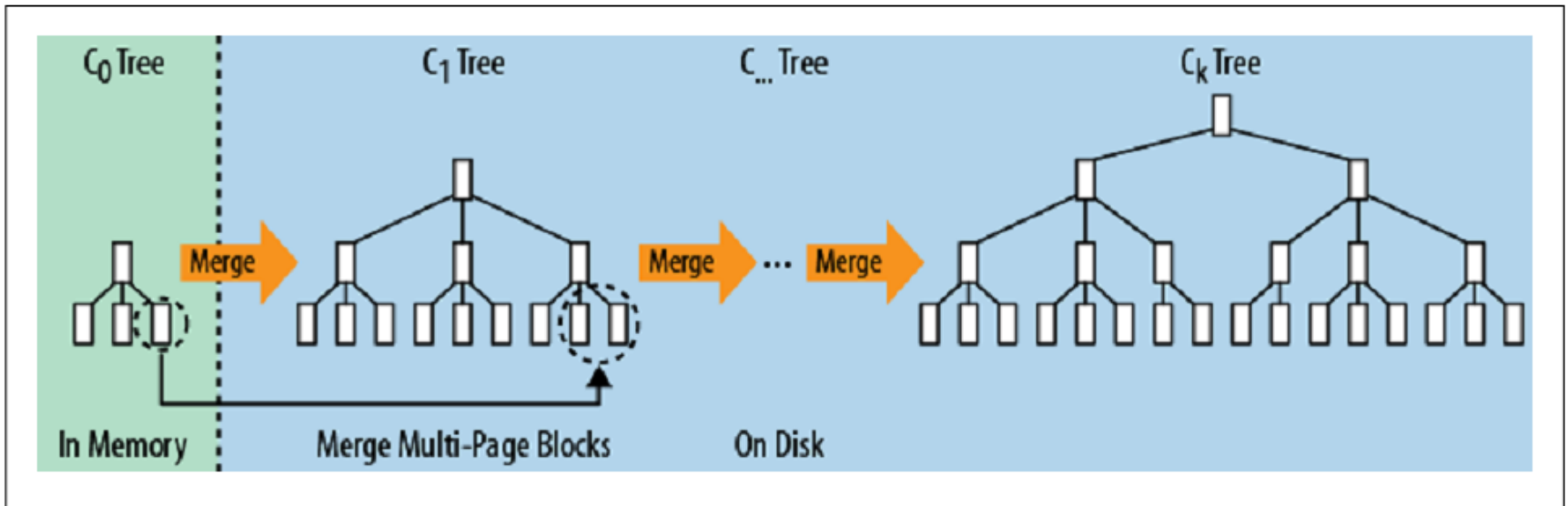
HBase架构原理-核心知识点

HBase核心知识点

- LSM
- HFile + WAL + Replication
- RIT
- Load Balance
- Split & Compaction
- GET + PUT + SCAN

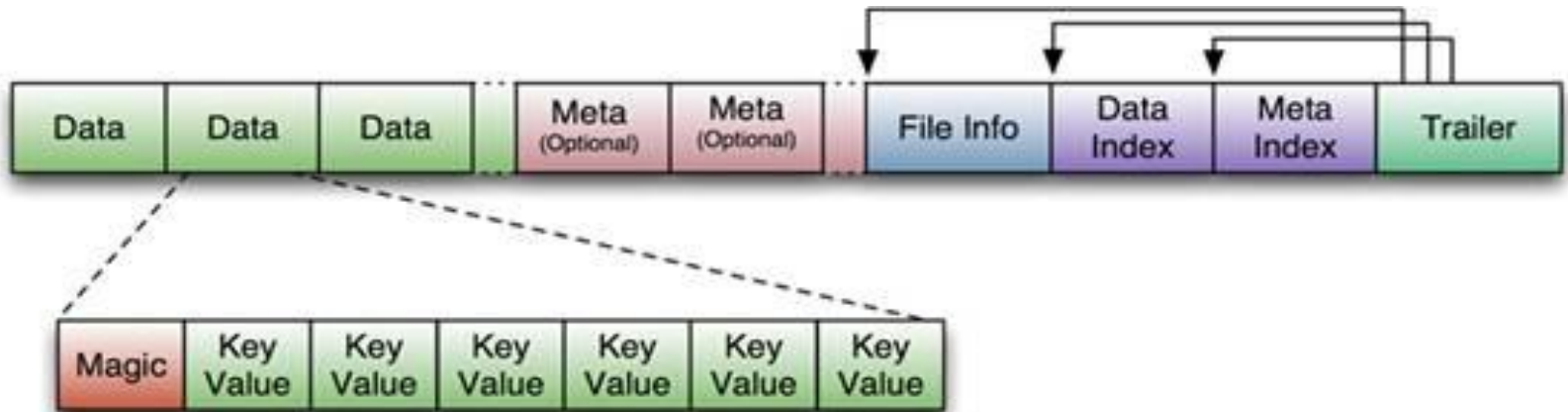
HBase核心知识点

- LSM - Log Structured Merge Tree
 - 顺序存储、内存中、flush磁盘
 - 读写独立、保证一致的写效率



HBase核心知识点

- HFILE: 附带索引的数据格式



- HBase中KeyValue数据的存储格式，HFile是Hadoop的二进制格式文件，实际上StoreFile就是对HFile做了轻量级包装，即StoreFile底层就是HFile

HBase核心知识点

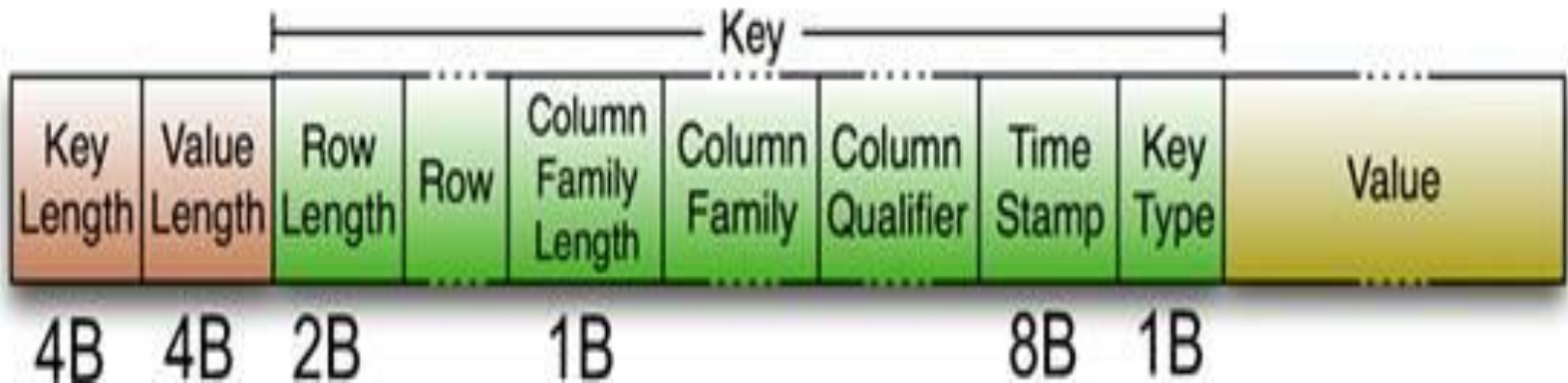
- HFile文件是不定长的，长度固定的只有其中的两块：Trailer和FileInfo。
- Trailer中有指针指向其他数据块的起始点
- File Info中记录了文件的一些Meta信息，例如：AVG_KEY_LEN, AVG_VALUE_LEN, LAST_KEY, COMPARATOR, MAX_SEQ_ID_KEY等
- Data Index和Meta Index块记录了每个Data块和Meta块的起始点。

HBase核心知识点

- 每个Data块除了开头的Magic以外就是一个个KeyValue对拼接而成
- Magic内容就是一些随机数字，目的是防止数据损坏。后面会详细介绍每个KeyValue对的内部构造

HBase核心知识点

➤ HFile里面的每个KeyValue对就是一个简单的byte数组。但是这个byte数组里面包含了很多项，并且有固定的结构。我们来看看里面的具体结构：



HBase核心知识点

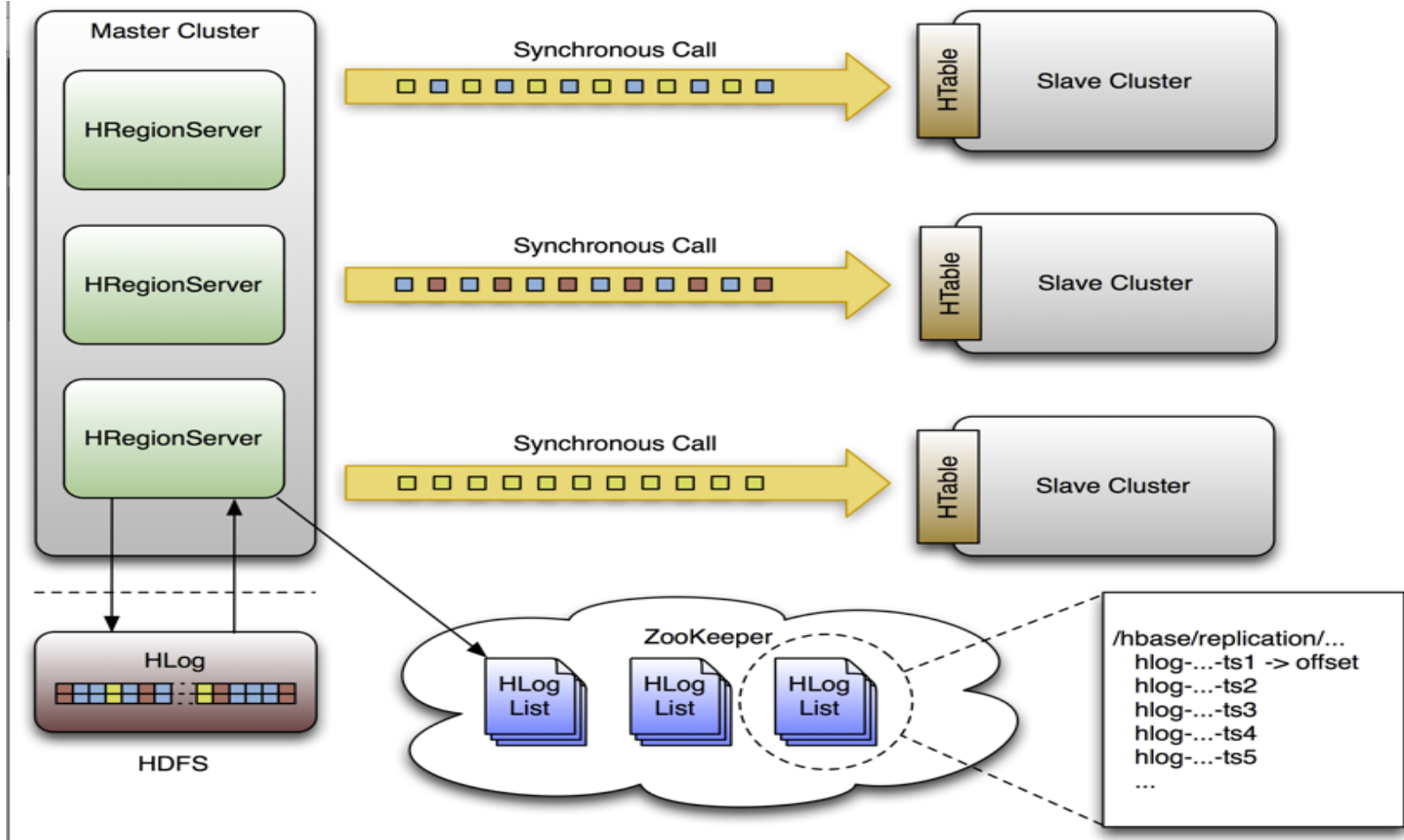
- 开始是两个固定长度的数值，分别表示Key的长度和Value的长度。
- Key: 开始是固定长度的数值，表示RowKey的长度，RowKey，然后固定长度的数值，表示Family长度，然后是Family，Qualifier，两个固定长度值，表示Time Stamp和Key Type (Put/Delete)。
- Value部分没有这么复杂的结构，就是纯粹的二进制数据了。

HBase核心知识点

- WAL: 数据持久化
 - Write Ahead Log
 - Write Log at first, then write data

HBase核心知识点

- Replication



HBase核心知识点

- RIT : Region In Transition
 - opening region
 - closing region
 - splitting region



HBase核心知识点

Table 8-1. Possible states of a region

State	Description
Offline	The region is offline.
Pending Open	A request to open the region was sent to the server.
Opening	The server has started opening the region.
Open	The region is open and fully operational.
Pending Close	A request to close the region has been sent to the server.
Closing	The server is in the process of closing the region.
Closed	The region is closed.
Splitting	The server started splitting the region.
Split	The region has been split by the server.

HBase核心知识点

- Load Balance
 - master每5分钟（默认）
hbase.balancer.period设定
 - balance_switch 是shell命令人工控制load
balance

HBase核心知识点

- Split

- Split : 平衡数据

- hbase.hregion.max.filesize

- 1. 默认10GB , 根据rowkey一分为二

- 当HStore中任何一个HStoreFile > 10GB (默认值) 时 , region会触发split操作 , 根据rowkey一分为二

HBase核心知识点

- Compaction :
 - minor compact
 - major compact

HBase核心知识点

- minor compact
 - (1) 如果需要合并的文件数 \geq `hbase.hstore.compactionThreshold` (默认3) , 继续
 - (2) $\text{min.size} < \text{filesize} \ \&\& \ \text{filesize} \leq \text{sum}(\text{比它小的文件}) * \text{hbase.hstore.compaction.ratio} \ \&\& \ \text{filesize} < \text{max.size}$

HBase核心知识点

则将文件按照从老到新的顺序添加到要合并的list中

➤ (3) list.size <=

hbase.hstore.compaction.max (10)

➤ (4) 合并文件

HBase核心知识点

- major compact
 - 当进行完一次major compaction以后所有的store都只有一个storefiles，这对查询性能有很大提升。注意：major compaction将会在一个运行中的系统中重写全部store的数据，不提倡自动运行；major compactions 在大型系统中通常手动进行。

HBase核心知识点

- 在hbase shell中使用 major_compact命令
 - 如果用户调用了major_compact命令或者majorCompact()API调用，都会强制major compaction运行。否则，服务端会首先检查是否该进行major compaction，通过查看距离上次运行是否满足一定时间，比如是否达到24小时。

HBase高级应用

HBase高级应用简单介绍

- bulk load
- OpenTSDB
- 优化
- key design
- 二级索引
- Coprocessors
- bloomfilter
- 版本

HBase高级应用简单介绍

Bulk load

- MapReduce实现
- `hadoop jar *.jar importcsv -`
`Dimporttsv.columns=a,b,c <tablename>`
`<inputdir>`

HBase高级应用简单介绍

优化

- read
- write
- hbase-site.xml
- JVM OPTS
- split & compact

HBase高级应用简单介绍

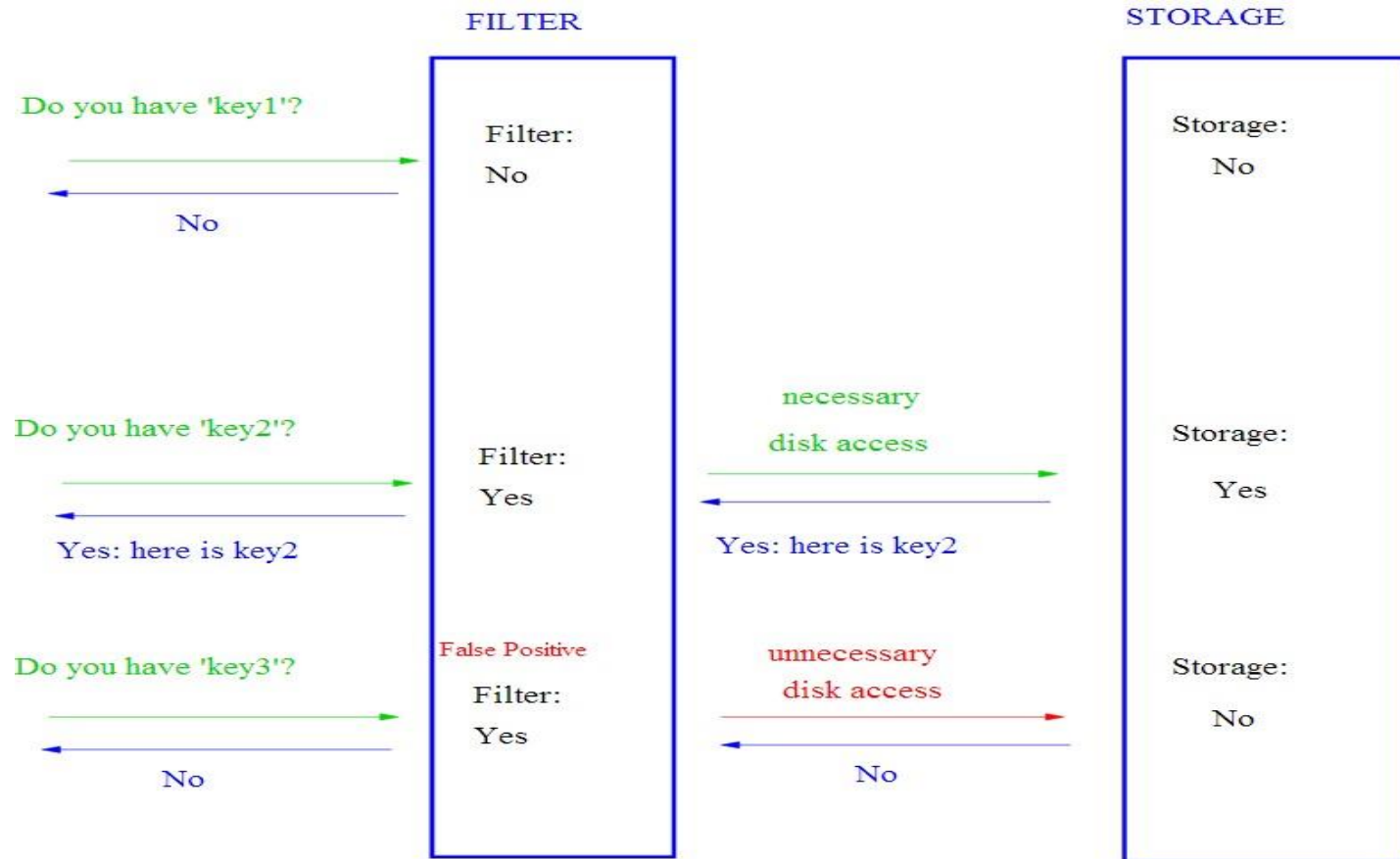
- key design
- 根据业务需求定义
 - uid
 - uid_ts
 - ts

HBase高级应用简单介绍

- 二级索引
- 不用于RDBMS索引
- rowkey为一级索引
- 核心是倒排表
- MapReduce、server-side、coprocessor

HBase高级应用简单介绍

- BloomFilter



安装部署

安装部署-预装

- LINUX
- SUN-JDK1.6
- Hadoop-1.0.4
- 2台服务器（虚拟机）

安装部署-install

- Hbase-0.94.8 download

<http://mirrors.tuna.tsinghua.edu.cn/apache/hbase/>

- mv hbase-0.94.8.tar.gz /opt/modules/
- tar -zxvf hbase-0.94.8.tar.gz
- cd hbase-0.94.8

安装部署-install

- hostname
- iptables
- jps

安装部署-deploy

In hbase-0.94.8/conf

- regionservers -- region hostname
- hbase-site.xml -- hbase configuration
- hbase-env.sh -- environment

安装部署-deploy

```
<property>
  <name>hbase.rootdir</name> <value>hdfs://hbase-
    master:9000/hbase</value>
</property>
<property>
  <name>hbase.cluster.distributed</name> <value>true</value>
</property>
<property>
  <name>hbase.master</name> <value>hbase-master:60000</value>
</property>
<property>
  <name>hbase.zookeeper.quorum</name> <value>zk0</value>
</property>
<property>
  <name>dfs.support.append</name> <value>true</value>
</property>
<property>
  <name>hbase.tmp.dir</name> <value>/tmp/hbase-${user.name}</value>
</property>
```

安装部署-deploy

```
export JAVA_HOME=/usr/java/default
```

```
export HBASE_HEAPSIZE=1000
```

```
export HBASE_MANAGES_ZK=true
```

安装部署-run

- 使用hadoop用户：su hadoop
- 查看java进程 /usr/java/default/bin/jps
- 首先启动zookeeper：

bin/hbase-daemon.sh start zookeeper

- 启动master：
- bin/hbase-daemon.sh start master

安装部署-run

- 启动regionserver :

`bin/hbase-daemon.sh start regionserver`

- 启动thrift server :

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

`threadpool -m200 -w 2000 -q 2000`

安装部署-run

- 检查是否启动正常
- jps和tail日志
- web ui

master机器浏览器中：

127.0.0.1:60010

- hbase shell

master机器hbase目录：

bin/hbase shell

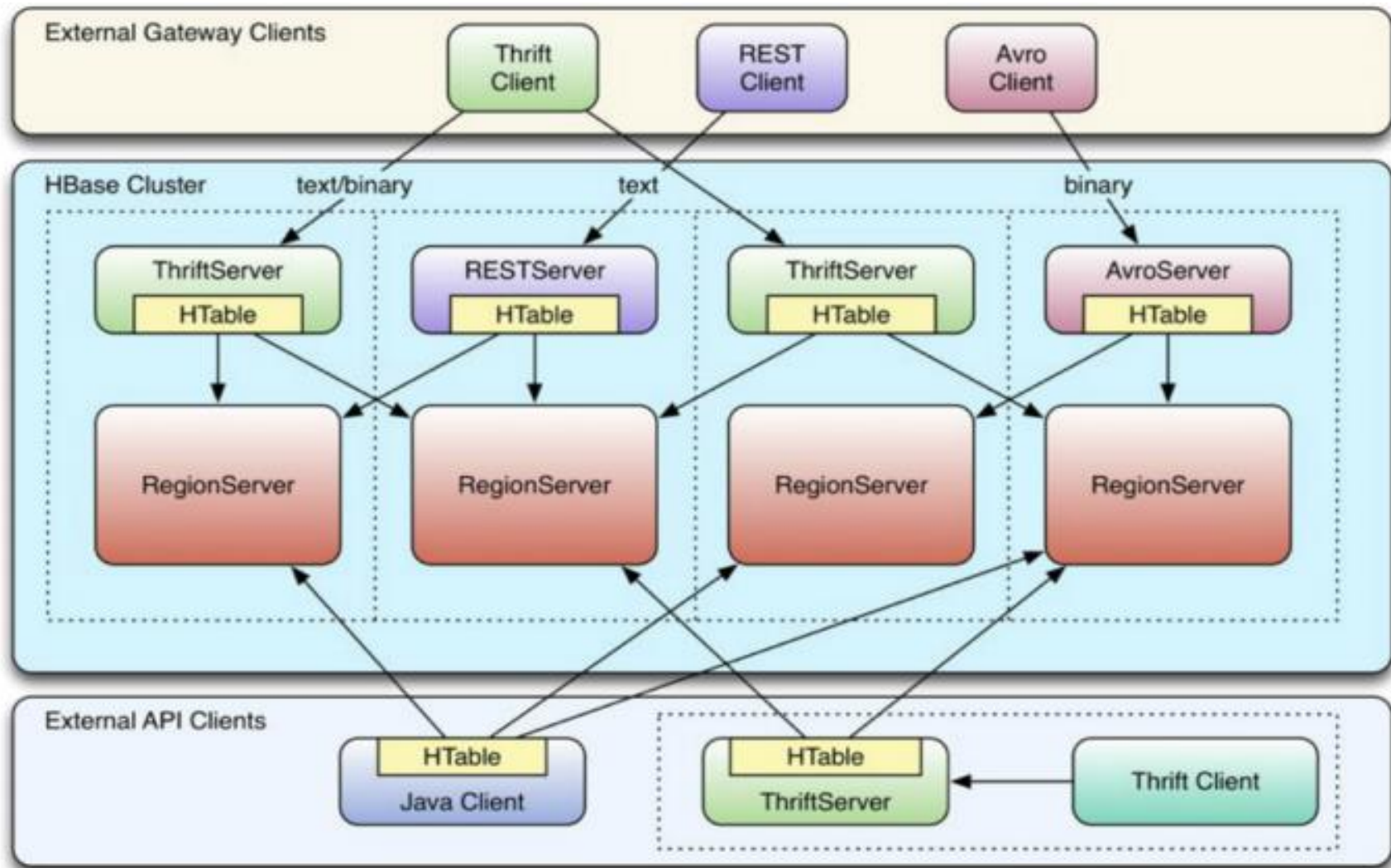
list

安装部署-启动顺序

- 启动Hadoop -> 启动ZooKeeper集群 -> 启动HBase -> 停止HBase -> 停止ZooKeeper集群 -> 停止Hadoop
- 对于hbase启动：一般先启动hbase master，然后是regionserver，他们之间没有严格的先后关系

开发设计

支持多种访问模式



常用接口

- HBase shell
- Native Java
- Thrift
- Batch clients : Mapreduce/hive/pig

常用接口-shell&native java

- Shell command-
list/create/scan/get/delete...

- Native java

public Configuration config;

public HTable table;

public HBaseAdmin admin;

开发设计

- Get : 读操作类
- Result get(Get get) throws IOException
- Result[] get(List<Get> gets) throws
IOException
- boolean exists(Get get) throws IOException

开发设计

- `Configuration conf = HBaseConfiguration.create();`
- `HTable table = new HTable(conf, "testtable");`
- `Get get = new Get(Bytes.toBytes("row1"));`
- `get.addColumn(Bytes.toBytes("colfam1"),
Bytes.toBytes("qual1"));`
- `Result result = table.get(get);`
- `byte[] val = result.getValue(Bytes.toBytes("colfam1"),
Bytes.toBytes("qual1"));`
- `System.out.println("Value: " + Bytes.toString(val));`

开发设计

- Put : 写操作类
- Put(byte[] row)
- Put(byte[] row, long ts)

开发设计

- `Configuration conf = HBaseConfiguration.create();`
- `HTable table = new HTable(conf, "testtable");`
- `Put put = new Put(Bytes.toBytes("row1 "));`
- `put.add(Bytes.toBytes("colfam1 "),
Bytes.toBytes("qual1 "),`
- `Bytes.toBytes("val1 "));`
- `table.put(put);`

开发设计

- Scan : 扫描 (读)
- 全表扫描类
- Scan()
- Scan(byte[] startRow, Filter filter)
- Scan(byte[] startRow)
- Scan(byte[] startRow, byte[] stopRow)

开发设计

- `Scan scan1 = new Scan();`
- `ResultScanner scanner1 = table.getScanner(scan1);`
- `for (Result res : scanner1) {`
 `System.out.println(res);`
`}`
- `scanner1.close();`

常用接口-thrift

- 支持语言-Java, C++, Perl, PHP, Python, Ruby...
- 服务端 : bin/hbase thrift start
- thrift & thrift2

常用接口-thrift

- 客户端 : \$ cp -r
\$HBASE_HOME/src/main/resources/org/apache/hadoop/hbase/thrift ~/thrift_src
- \$ cd thrift_src/
- \$ thrift -gen **java** Hbase.thrift
- \$HBASE_HOME/src/examples/thrift/DemoClient.php

常用接口-thrift

- Host=*.~*.~*(ip); port = 9090;
- 关键代码：

```
TTransport transport = new TSocket(host, port);
```

```
TProtocol protocol = new
```

```
    TBinaryProtocol(transport, true, true);
```

```
Hbase.Client client = new Hbase.Client(protocol);
```

```
transport.open();
```

- Mutation - wrapper of PUT
- client.mutateRow() – commit PUT

常用接口-Mapreduce/hive/pig

- Mapreduce :
 - LzoTextInputFormat/TextInputFormat
 - TableInputFormat/TableOutputFormat
- Hive/pig
- ...

HBase的Java编程

HBase Java编程接口概述

HBaseConfiguration

HBaseConfiguration是每一个hbase client都会使用到的对象，它代表的是HBase配置信息。

默认的构造方式会尝试从hbase-default.xml和hbase-site.xml中读取配置。如果classpath没有这两个文件，就需要你自己设置配置。

```
Configuration HBASE_CONFIG = new Configuration();
```

```
HBASE_CONFIG.set("hbase.zookeeper.quorum", "zkServer");
```

```
HBASE_CONFIG.set("hbase.zookeeper.property.clientPort",  
"2181");
```

```
HBaseConfiguration cfg = new  
HBaseConfiguration(HBASE_CONFIG);
```


HBase Java编程接口概述 创建表

- 创建表是通过HBaseAdmin对象来操作的。HBaseAdmin负责表的META信息处理。HBaseAdmin提供了createTable这个方法：`public void createTable(HTableDescriptor desc)`
- HTableDescriptor 代表的是表的schema
- HColumnDescriptor 代表的是column的schema
 - `HBaseAdmin hAdmin = new HBaseAdmin(hbaseConfig);`
 - `HTableDescriptor t = new HTableDescriptor(tableName);`
 - `t.addFamily(new HColumnDescriptor("f1"));`
 - `t.addFamily(new HColumnDescriptor("f2"));`
 - `t.addFamily(new HColumnDescriptor("f3"));`
 - `t.addFamily(new HColumnDescriptor("f4"));`
 - `hAdmin.createTable(t);`

HBase Java编程接口概述 删除表

删除表通过HBaseAdmin来操作，删除表之前首先要disable表。这是一个非常耗时的操作，所以不建议频繁删除表。
disableTable和deleteTable分别用来disable和delete表

```
HBaseAdmin hAdmin = new HBaseAdmin(hbaseConf);
```

```
if (hAdmin.tableExists(tableName)) {  
    hAdmin.disableTable(tableName);  
    hAdmin.deleteTable(tableName);  
}
```

HBase Java编程接口概述 删除表

删除表通过HBaseAdmin来操作，删除表之前首先要disable表。这是一个非常耗时的操作，所以不建议频繁删除表。
disableTable和deleteTable分别用来disable和delete表

```
HBaseAdmin hAdmin = new HBaseAdmin(hbaseConf);
```

```
if (hAdmin.tableExists(tableName)) {  
    hAdmin.disableTable(tableName);  
    hAdmin.deleteTable(tableName);  
}
```

HBase Java编程接口 查询数据

- 查询分为单条随机查询和批量查询。单条查询是通过rowkey在table中查询某一行的数据。HTable提供了get方法来完成单条查询。
- 批量查询是通过制定一段rowkey的范围来查询。HTable提供了个getScanner方法来 完成批量查询。
 - Scan s = new Scan();
 - s.setMaxVersions();
 - ResultScanner ss = table.getScanner(s);
 - for(Result r:ss){
 - System.out.println(new String(r.getRow()));
 - for(KeyValue kv:r.raw()){
 - System.out.println(new String(kv.getColumn()));
 - }
 - }



HBase运营维护

HBase监控

- Master WEB UI(自帶)
- Ganglia
- OpenTSDB
- Ella

Master WEB UI



Master: localhost:60000

[Local logs](#), [Thread Dump](#), [Log Level](#)

Master Attributes

Attribute Name	Value	Description
HBase Version	0.91.0-SNAPSHOT, r1127782	HBase version and svn revision
HBase Compiled	Thu May 26 10:28:47 CEST 2011, larsgeorge	When HBase version was compiled and by whom
Hadoop Version	0.20-append-r1057313, r1057313	Hadoop version and svn revision
Hadoop Compiled	Wed Feb 9 22:25:52 PST 2011, Stack	When Hadoop version was compiled and by whom
HBase Root Directory	hdfs://localhost:8020/hbase	Location of HBase home directory
HBase Cluster ID	698e057d-78ac-4d01-8db9-3cec937bc619	Unique identifier generated for each HBase cluster
Load average	4	Average number of regions per regionserver. Naive computation.
Zookeeper Quorum	localhost:2181	Addresses of all registered ZK servers. For more, see zk.dump .

Currently running tasks

No tasks currently running on this node.

Catalog Tables

Table	Description
-ROOT-	The -ROOT- table holds references to all .META. regions.
.META.	The .META. table holds references to all User Table regions

User Tables

3 table(s) in set.

Table	Description
testtable	{NAME => 'testtable', FAMILIES => [{NAME => 'colfam1', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COMPRESSION => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}]}
user	{NAME => 'user', DEFERRED_LOG_FLUSH => 'false', READONLY => 'false', MEMSTORE_FLUSHSIZE => '67108864', MAX_FILESIZE => '268435456', FAMILIES => [{NAME => 'data', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', COMPRESSION => 'NONE', VERSIONS => '3', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}]}
usertable	{NAME => 'usertable', FAMILIES => [{NAME => 'family', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', TTL => '2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}]}

Region Servers

	Address	Start Code	Load
	localhost:60030	1306411472676localhost:60020,1306411472676	requests=0, regions=4, usedHeap=53, maxHeap=987
Total:	servers: 1		requests=0, regions=4

Load is requests per second and count of regions loaded

Regions in Transition

No regions in transition.

Monitor

- Ganglia

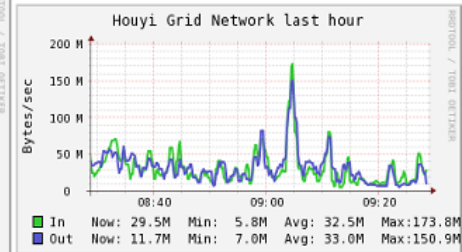
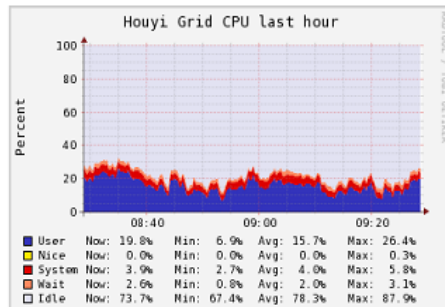
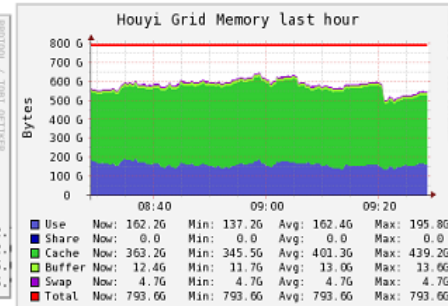
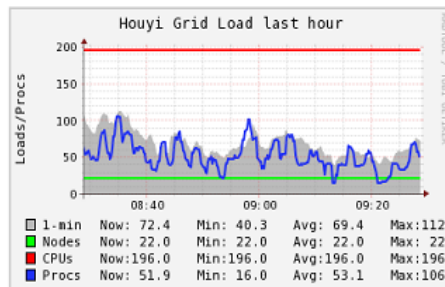
Houyi Grid (5 sources) (tree view)

CPU's Total: **196**
Hosts up: **22**
Hosts down: **0**

Current Load Avg (15, 5, 1m):
30%, 31%, 37%

Avg Utilization (last hour):
35%

Localtime:
2013-04-25 09:28

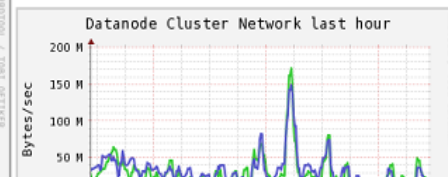
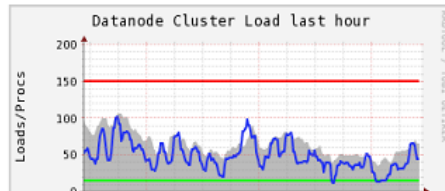


Datanode (physical view)

CPU's Total: **152**
Hosts up: **16**
Hosts down: **0**

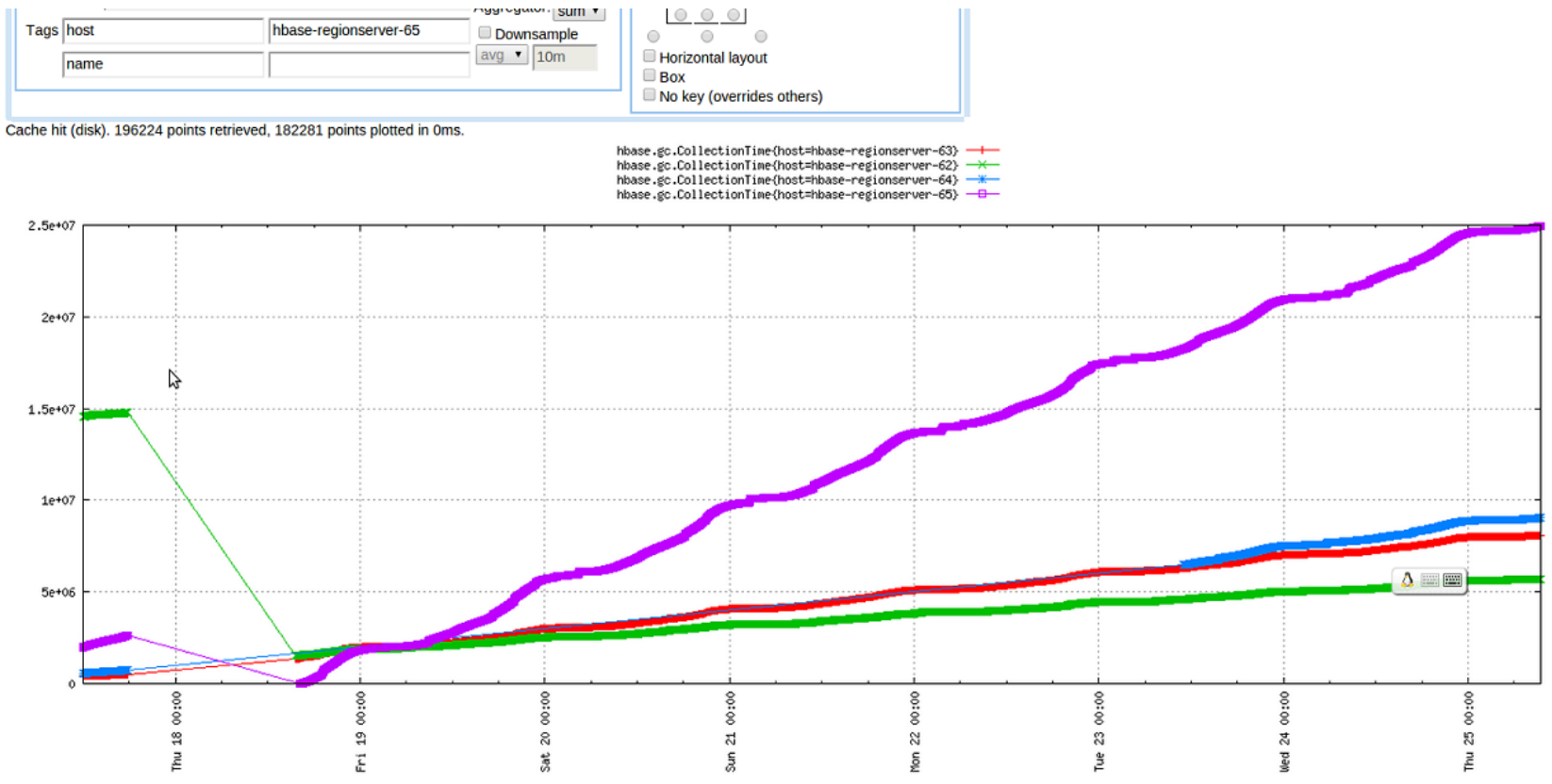
Current Load Avg (15, 5, 1m):
36%, 36%, 42%

Avg Utilization (last hour):
43%



Monitor

- OpenTSDB



Monitor

- Ella : Table、Region级别监控

Ella: A Watchdog on HBase

APACHE HBASE

表访问量监控
Hot Region监控
Server监控
Zookeeper监控

今日数据						
Table Name	Write Count	Read Count	Total Count	Write TPS	Read TPS	Total TPS
movie_fav_visitor_list	0	4	4	0	0	0
message_visitor	17299	29295	46594	0	0	0
user_behavior_attribute_noregistered_mid_uid_index	2151413895	2146293904	4297707799	0	0	0
tsdb	15836759	303090	16139849	2	0	2
user_behavior_attribute_noregistered_index	36	279	315	0	0	0
user_behavior_attribute	1134153963	101869131	1236023094	207	23	230
user_behavior_attribute_noregistered	871755156	7381025064	8252780220	0	750	750
tsdb-uid	0	1637	1637	0	0	0
movie_fav_user_list	7323829	2971450	10295279	0	0	0
message_user	204509566	93197559	297707125	12	80	93
META	1400	5307851	5309251	0	0	0
-ROOT-	2	24368	24370	0	0	0

The red tables are being accessed!

Monitor

• PhpHBaseAdmin

phpHBaseAdmin

Tables ▾

Monitor ▾

System ▾

message_user

message_visitor

movie_fav_user_list

movie_fav_visitor_list

test

user_behavior_attribut...

user_behavior_attribut...


user_behavior_attribut...

user_behavior_attribut...

user_test

1

phpbaseadmin



Attribute Name	Value
ServerName	hbase-master ,60000
version	0.94.0, r1332822
zookeeper_quorum	hbase-regionserver-63:2181,hbase-regionserver-64:2181,hbase-regionserver-62:2181
DeadRegionServers	
AverageLoad	39.25
MasterStartTime	2013-08-09 10:33:29
live_regionservers	8
Coprocessors	

HBase Shell运维命令

- Group name: tools
- Commands: assign, balance_switch, balancer, close_region, compact, flush, hlog_roll, major_compact, move, split, unassign, zk_dump

HBase Shell

命令	命令含义	命令使用实例
assign	分配Region	assign 'region1'
balance_switch	启用或关闭负载均衡器，返回结果是当前均衡器状态	balance_switch truebalance_switch false
balancer	触发集群负载均衡器。如果成功运行返回	balancer
close_region	关闭某个Region	close_region 'REGIONNAME'close_region 'REGIONNAME', 'SERVER_NAME'
compact	合并表或Region	compact 't1'compact 'r1', 'c1'compact 't1', 'c1'
flush	Flush	flush 'TABLENAME'flush 'REGIONNAME'
hlog_roll	HLog	hlog_roll 'REGIONSERVERNAME'
major_compact	大合并表或Region	major_compact 't1'major_compact 'r1', 'c1'major_compact 't1', 'c1'
move	移动Region	move 'ENCODED_REGIONNAME'move 'ENCODED_REGIONNAME', 'SERVER_NAME'
split	分裂表或Region	split 'tableName'split 'regionName' # format: 'tableName,startKey,id'split 'tableName', 'splitKey'split 'regionName', 'splitKey'
unassign	解除指定某个Region	unassign 'REGIONNAME'unassign 'REGIONNAME', true
zk_dump	打印输出	zk_dump

参数配置

HBase Shell

In hbase-0.94.8/conf

- regionservers -- region hostname
- hbase-site.xml -- hbase configuration
- hbase-env.sh -- environment

regionservers

配置所有的RS的hostname列表

(1) 可以不配置

(2) 使用start-hbase.sh命令必须配置

regionserver

示例：

```
cat conf/regionserver
```

hbase-rs1

hbase-rs2

hbase-rs3

hbase-site.xml

主要的配置参数文件，包含所有hbase相关的
细节参数：HDFS、端口、日志、客户端、
RegionServer、Master、Region、Store
级别、zk、安全等等

详细见 `src/main/resources/hbase-
default.xml`文件，课堂讲解

hbase-site.xml

主要的配置参数文件，包含所有hbase相关的
细节参数：HDFS、端口、日志、客户端、
RegionServer、Master、Region、Store
级别、zk、安全等等

hbase-env.sh

环境变量配置文件，配置：

JAVA_HOME

HEAP_SIZE

JAVA GC

MANAGE ZK

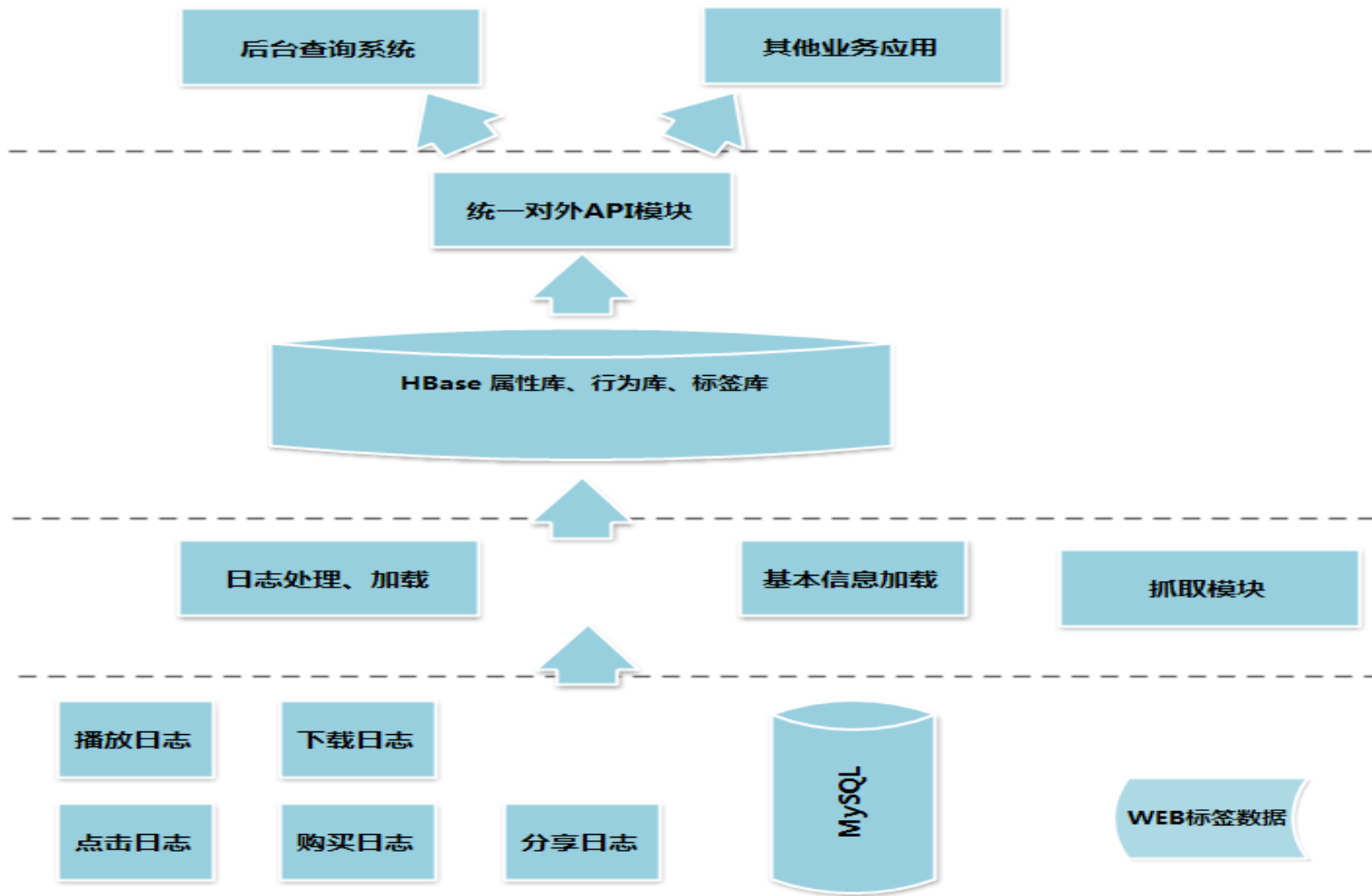
参照文件课堂讲解

案例剖析

案例剖析

- 推荐服务
- 实时计算系统底层存储
- 特殊报表统计需求
- 数据质量监控系统

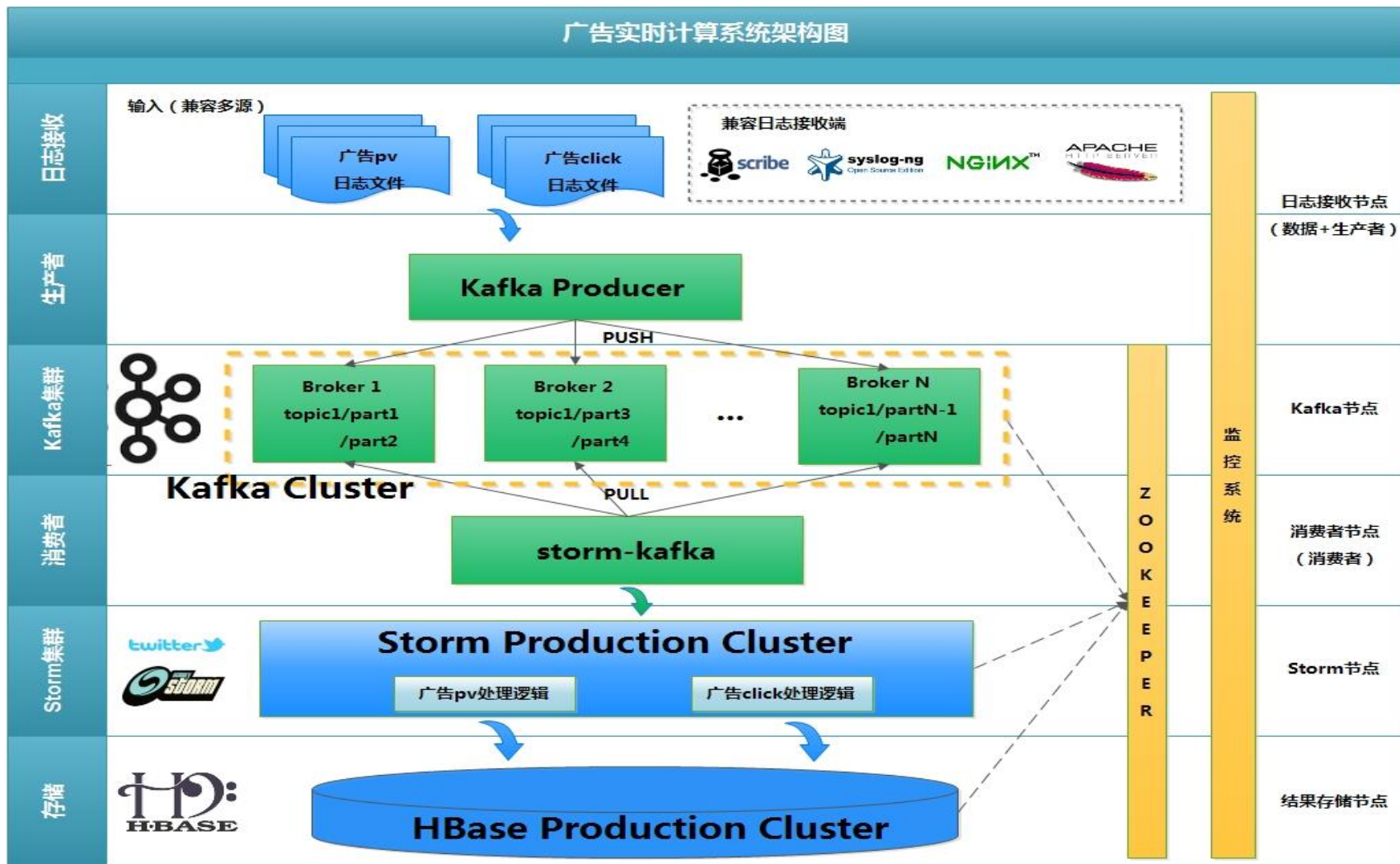
用户属性库（推荐、广告）



用户属性库（推荐、广告）

主键、列簇	字段名称	字段解释	字段值举例
rowkey	表主键	拼接主键	attr:movt_area_
timestamp	时间戳	版本控制	
cf1	列簇		
	rk	用户	50396612-AB2B-F5B78D8B10A2
	uid_cnt	统计字段	253458
cf2	扩展列簇		

实时计算系统



实时计算系统

- HBase作为最终数据存储系统
- HBase提供对外检索服务
- 检索服务之前可以添加对应缓存层

总结与提问

