

— 千 锋 强 力 推 出 —

逆战班

停 课 不 停 学

品 质 不 打 折

鼓励全国学子在疫情中坚持学习



01 HBase在Hadoop
生态圈中的位置

02 HBase体系结构之
物理模型

03 HBase体系结构之
存储数据的过程

04 -ROOT-和.META.

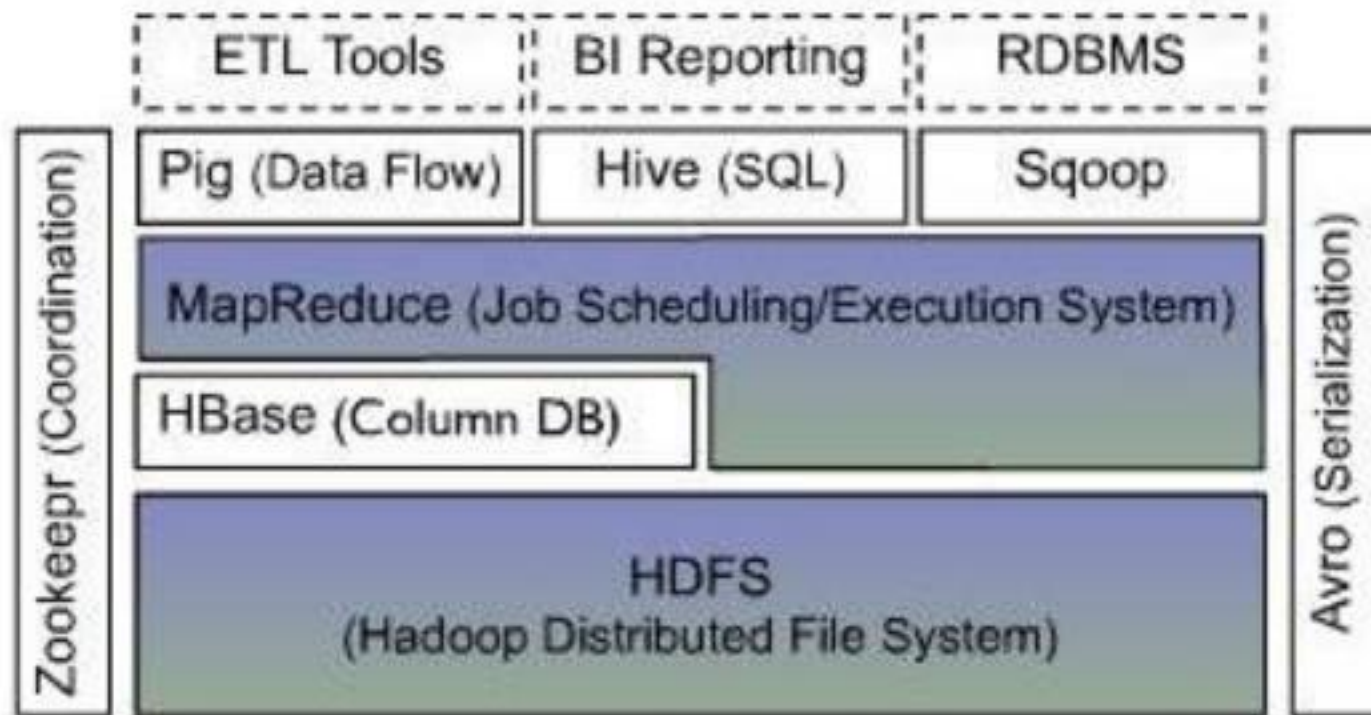
05 HBase相关面试题汇总

课程目标

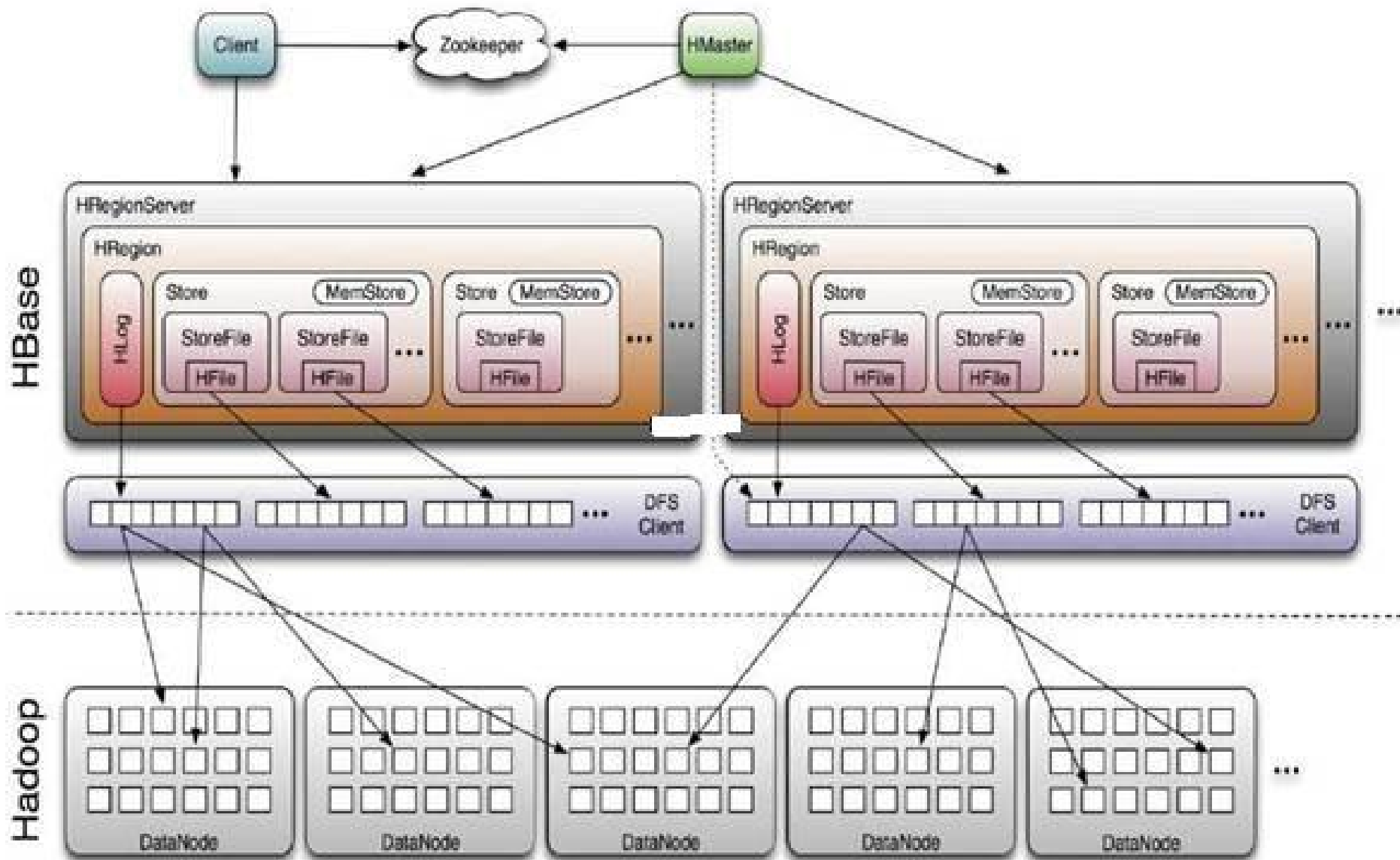
COURSE CONTENTS

HBase在Hadoop生态圈中的位置

The Hadoop Ecosystem



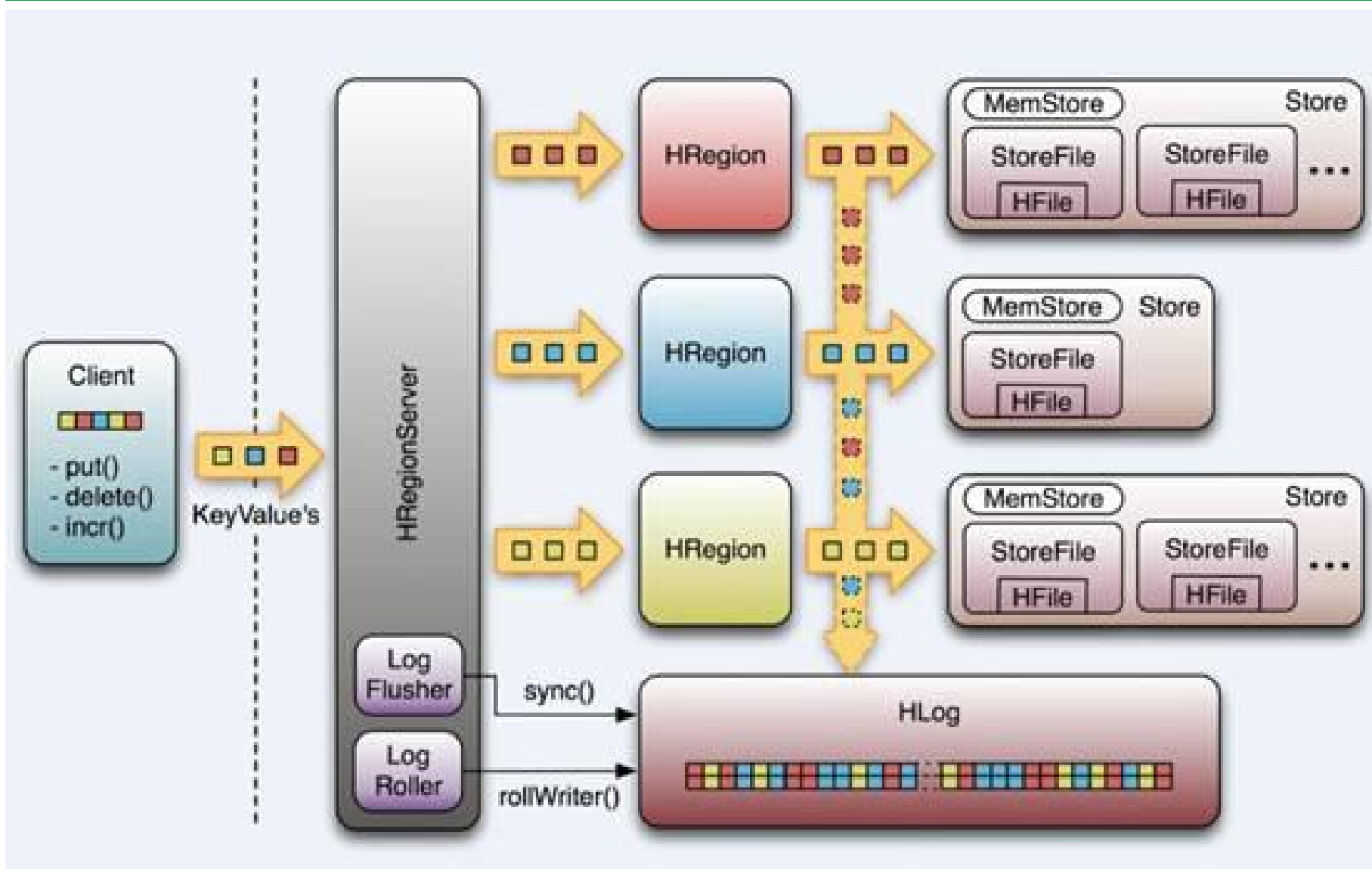
HBase体系结构之物理模型



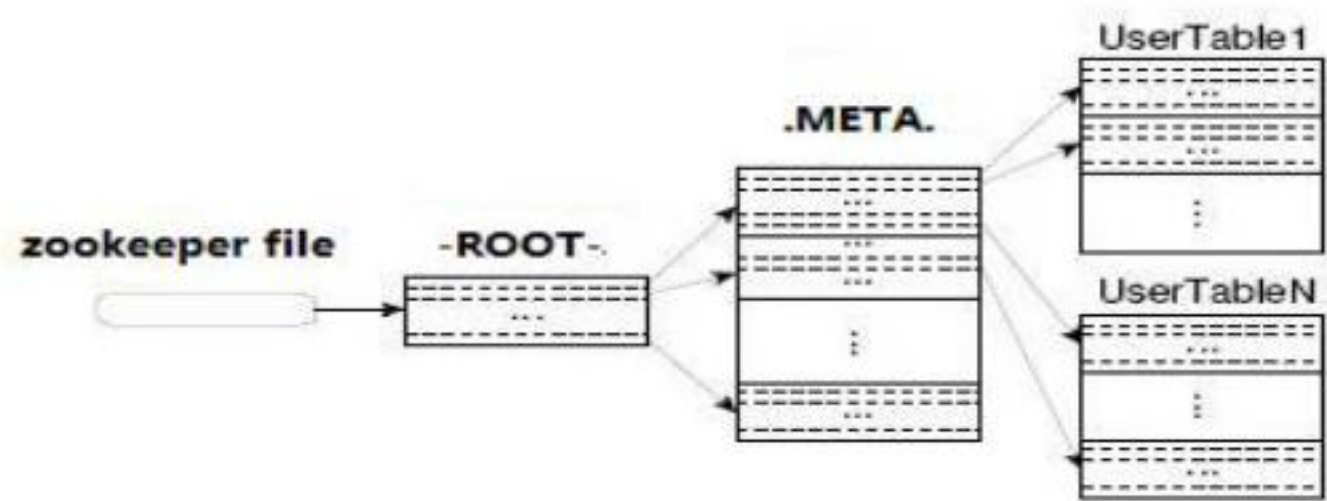
各组件间的数量关系：

- hmaster:hregionserver=1:n
- hregionserver:hregion=1:n
- hregionserver:hlog=1:1
- hregion:hstore=1:n
- store:memstore=1:1
- store:storefile=1:n
- storefile:hfile=1:1

HBase体系结构之存储数据的过程



-ROOT-和.META.



HBase中有两张特殊的Table，**-ROOT-和.META.**
-ROOT-：记录了.META.表的Region信息，-ROOT-只有一个region
.META.：记录了用户创建的表的Region信息，.META.可以有多个region

Zookeeper中记录了-ROOT-表的location

Client访问用户数据之前需要首先访问zookeeper，然后访问-ROOT-表，接着访问.META.表，最后才能找到用户数据的位置去访问

-ROOT-行记录结构

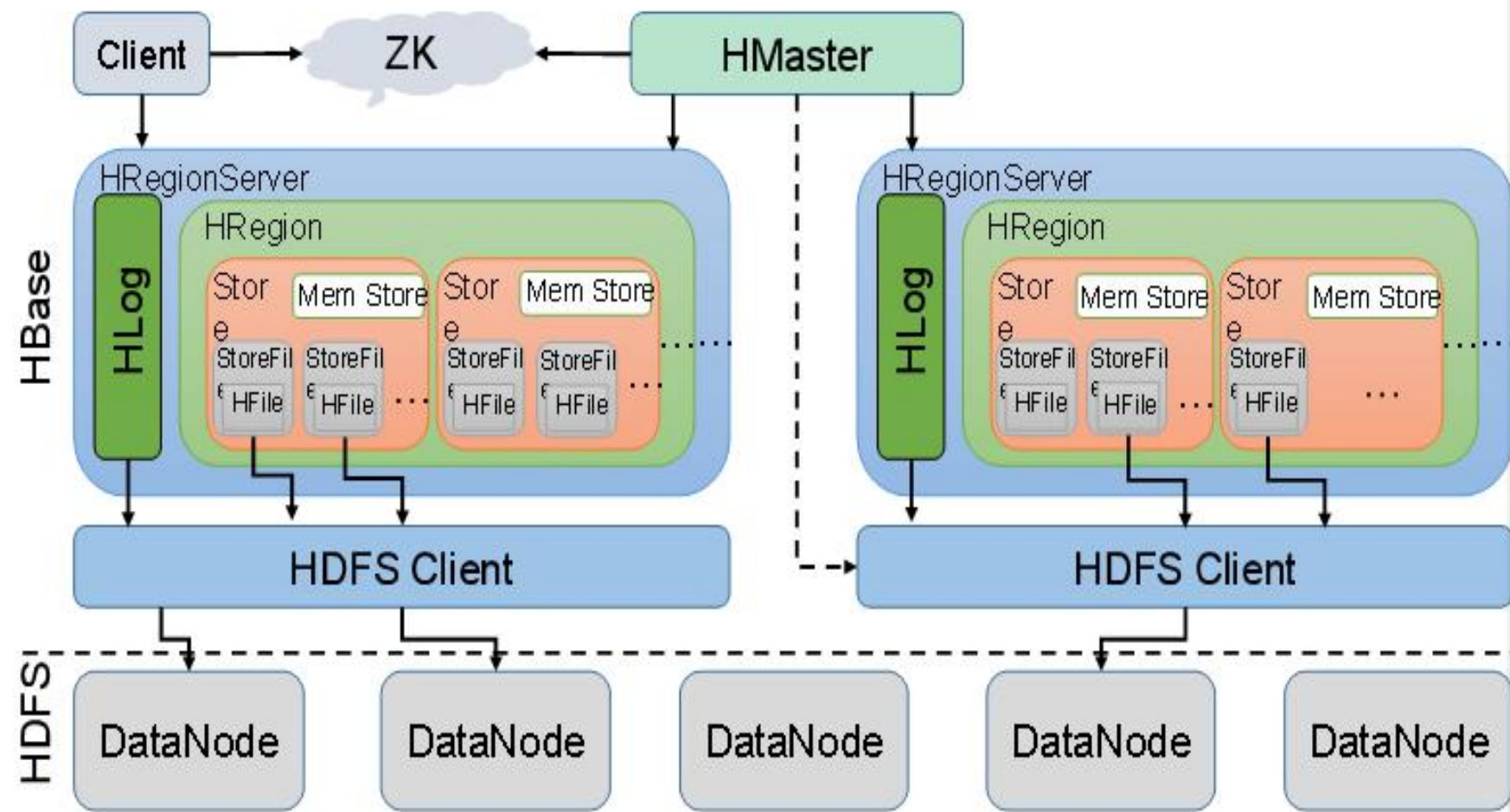
RowKey	info			historian
	regioninfo	server	server startcode	
.META.,Table1,0,12345678,12657843		RS1		
.META.,Table2,30000,12348765,12438675		RS2		

.META.行记录结构 →

RowKey	info			historian
	regioninfo	server	server startcode	
Table1,RK0,12345678		RS1		
Table1, RK10000,12345687		RS2		
Table1, RK20000,12346578		RS3		
...
Table2, RK0,12345678		RS1		
Table2, RK30000,12348765		RS2		

HBase面试题汇总

1、请绘制出HBase的存储结构。



HLog与HRegionServer一一对应

HLog持久化在HDFS之上, HLog存储位置查看→

`hadoop fs -ls /hbase/WALs`

如果HLog已经失效(所有之前的写入MemStore已经持久化在HDFS),HLog存在于HDFS之上的文件会从/hbase/.logs转移至/hbase/oldWALs, oldlogs会删除, HLog的生命周期结束.

向HBase Put数据时通过HBaseClient-->连接ZooKeeper--->-ROOT--->.META.-->RegionServer-->Region:

HBase相关面试题汇总

2、请阐述HBase中rowkey设计的原则。

- 1) rowkey长度原则
- 2) rowkey散列原则
- 3) rowkey唯一原则

3、请阐述HBase中RowKey如何设计？

- (1) 生成随机数、hash、散列值
- (2) 字符串反转

4、请阐述Phoenix二级索引的原理。

(二级索引实验 (<https://blog.csdn.net/whdxjbw/article/details/81146440>)

→ 概述

全局索引是Phoenix的重要特性，合理的使用二级索引能降低查询延时，让集群资源得以充分利用。

→ 全局索引说明

全局索引的根本是通过单独的HBase表来存储数据表的索引数据。我们通过如下示例看索引数据和主表数据的关系。

→ 给特定的字段添加索引

建议在phoenix客户端下演示

语法1：给表添加单个索引（或是多个，采取同步的方式，适用场景是：表中数据量小，添加二级索引时，不需要对表进行额外的操作）

单个字段： `create index 索引表的名 on 表(字段1)；`

多个字段： `create index 索引表的名 on 表(字段1) include(字段2)；`

语法2：给多个字段添加索引（采用异步的方式，适用场景是：表中数据量庞大，为了不影响表正常的使用）

`create index 索引表的名 on 表(字段1) include(字段2) async;`

如： `create index myindex on us_pop(state) include(city) async;`

phoenix使用hbase命令激活异步建立的二级索引

`hbase org.apache.phoenix.mapreduce.index.IndexTool --schema default --data-table UP_POP --index-table MYINDEX --output-path ASYNC_IDX_HFILES`

HBase相关面试题汇总

4、请阐述Phoenix二级索引的原理（续）。

```
-- 创建数据表CREATE TABLE DATA_TABLE(  
  A VARCHAR PRIMARY KEY,  
  B VARCHAR,  
  C INTEGER ,  
  D INTEGER);  
  
-- 创建索引CREATE INDEX B_IDX ON DATA_TABLE(B)INCLUDE(C);  
  
-- 插入数据UPSERT INTO DATA_TABLE VALUES('A','B',1,2);
```

当写入数据到主表时，索引数据也会被同步到索引表中。索引表中的主键将会是索引列和数据表主键的组合值，include的列被存储在索引表的普通列中，其目的是让查询更加高效，只需要查询一次索引表就能够拿到数据，而不用去回查主表。其过程入下图



Phoenix表就是HBase表，而HBase Rowkey都是通过二进制数据的字典序排列存储，也就意味着Row key前缀匹配度越高就越容易排在一起。

HBase相关面试题汇总

4、请阐述Phoenix二级索引的原理（续）。

→ 全局索引设计

我们继续使用DATA_TABLE作为示例表，创建如下组合索引。之前我们已经提到索引表中的Row key是字典序存储的，什么样的查询适合这样的索引结构呢？

CREATE INDEX B_C_D_IDX ON DATA_TABLE(B,C,D);

所有字段条件以=操作符为例：

注：上表查询中and条件不一定要和索引组合字段顺序一致，可以任意组合。

在实际使用中我们也只推荐使用1~4，遵循前缀匹配原则，避免触发扫全表。5~7条件就要扫描全表数据才能过滤出来符合这些条件的数据，所以是极力不推荐的。

序号	查询	索引最高效程度
1	select * from DATA_TABLE where B='B' and C = 1 and D = 2	最高
2	select * from DATA_TABLE where B='B' and C = 1	较高
3	select * from DATA_TABLE where B='B' and D = 2	中
4	select * from DATA_TABLE where B='B'	中
5	select * from DATA_TABLE where C = 1 and D = 2	低
6	select * from DATA_TABLE where C = 1	低
7	select * from DATA_TABLE where D = 2	

HBase相关面试题汇总

4、请阐述Phoenix二级索引的原理（续）。

→ 其它：

对于order by字段或者group by字段仍然能够使用二级索引字段来加速查询。

尽量通过合理的设计数据表的主键规避建更多的索引表，因为索引表越多写放大越严重。

对索引表适当使用加盐特性能提升查询写入的性能，避免热点问题。

5、说说你对Phoenix的了解。

Phoenix是一个HBase的开源SQL引擎，它使开发者可以像访问普通数据库那样使用jdbc访问HBase。

→ 优点：

1. 支持SQL查询hbase，自动转换SQL为最佳并行scan语句
2. 将where子句交给过滤器处理，将聚合查询交给协处理器处理
3. 支持直接创建二级索引来提升非主键的查询性能
4. 跳过扫描过滤器来优化IN、Like、OR查询
5. 优化写操作

5、说说你对Phoenix的了解（续）。

→ 缺点：

1. 不支持事务、不支持复杂查询
2. 严格的版本限制，每个phoenix对应一个版本hbase
3. 与hbase强相关，可能导致元数据被破坏

→ 性能：

使用where子句时，phoenix几乎是即时返回，普通的hive on hbase则需要等待一段时间；

不使用where子句时，hive的延迟约是phoenix的3-40倍；

进行聚合计算时，性能远超Impala（CDH提供的hdfs查询引擎），约为30-70倍。（Impala是Cloudera公司主导开发的新型查询系统，它提供SQL语义，能查询存储在Hadoop的HDFS和HBase中的PB级大数据。）

6、Hbase和传统数据库的区别有哪些？

hbase是海量数据的分布式存储，响应时间为秒级，列式存储，二进制行键。

hbase最大的优势就是存储TB级别的数据量时增删改查速度几乎不变，而传统数据库则会随着数据量增加，性能成倍地下降。

hbase只有string类型；hbase只支持增删改查，没有join和子查询；

hbase元数据区分大小写