



# 尚硅谷大数据技术之HBase



列		personal_info			office_info		列族
	Row Key	name	city	phone	tel	address	
	row_key1	张三	北京	131*****	010-11111111	atguigu	
Row key	row_key11	李四	上海	132*****	010-11111111	atguigu	
	row_key2	王五	广州	159*****	010-11111111	atguigu	
	row_key3	赵六	深圳	187*****	010-11111111	atguigu	
	row_key4	横七	大连	134*****	010-11111111	atguigu	
	row_key5	竖八	重庆	139*****	010-11111111	atguigu	
	row_key6	金九	武汉	177*****	010-11111111	atguigu	
	row_key7	银十	保定	158*****	010-11111111	atguigu	



列		personal_info			office_info		列族
	Row Key	name	city	phone	tel	address	
Row key	row_key1	张三	北京	131*****	010-11111111	atguigu	Region
	row_key11	李四	上海	132*****	010-11111111	atguigu	
	row_key2	王五	广州	159*****	010-11111111	atguigu	
Region	row_key3	赵六	深圳	187*****	010-11111111	atguigu	Region
	row_key4	横七	大连	134*****	010-11111111	atguigu	
	row_key5	竖八	重庆	139*****	010-11111111	atguigu	
	row_key6	金九	武汉	177*****	010-11111111	atguigu	
	row_key7	银十	保定	158*****	010-11111111	atguigu	



store

列

列族

personal_info			
Row Key	name	city	phone
row_key1	张三	北京	131*****
row_key11	李四	上海	132*****
row_key2	王五	广州	159*****

office_info	
tel	address
010-11111111	atguigu
010-11111111	atguigu
010-11111111	atguigu

Row key

Region

row_key3	赵六	深圳	187*****	010-11111111	atguigu
row_key4	横七	大连	134*****	010-11111111	atguigu
row_key5	竖八	重庆	139*****	010-11111111	atguigu

row_key6	金九	武汉	177*****
row_key7	银十	保定	158*****

010-11111111	atguigu
010-11111111	atguigu



	personal_info		
Row Key	name	city	phone
row_key1	张三	北京	131*****
row_key11	李四	上海	132*****
row_key2	王五	广州	159*****

StoreFile

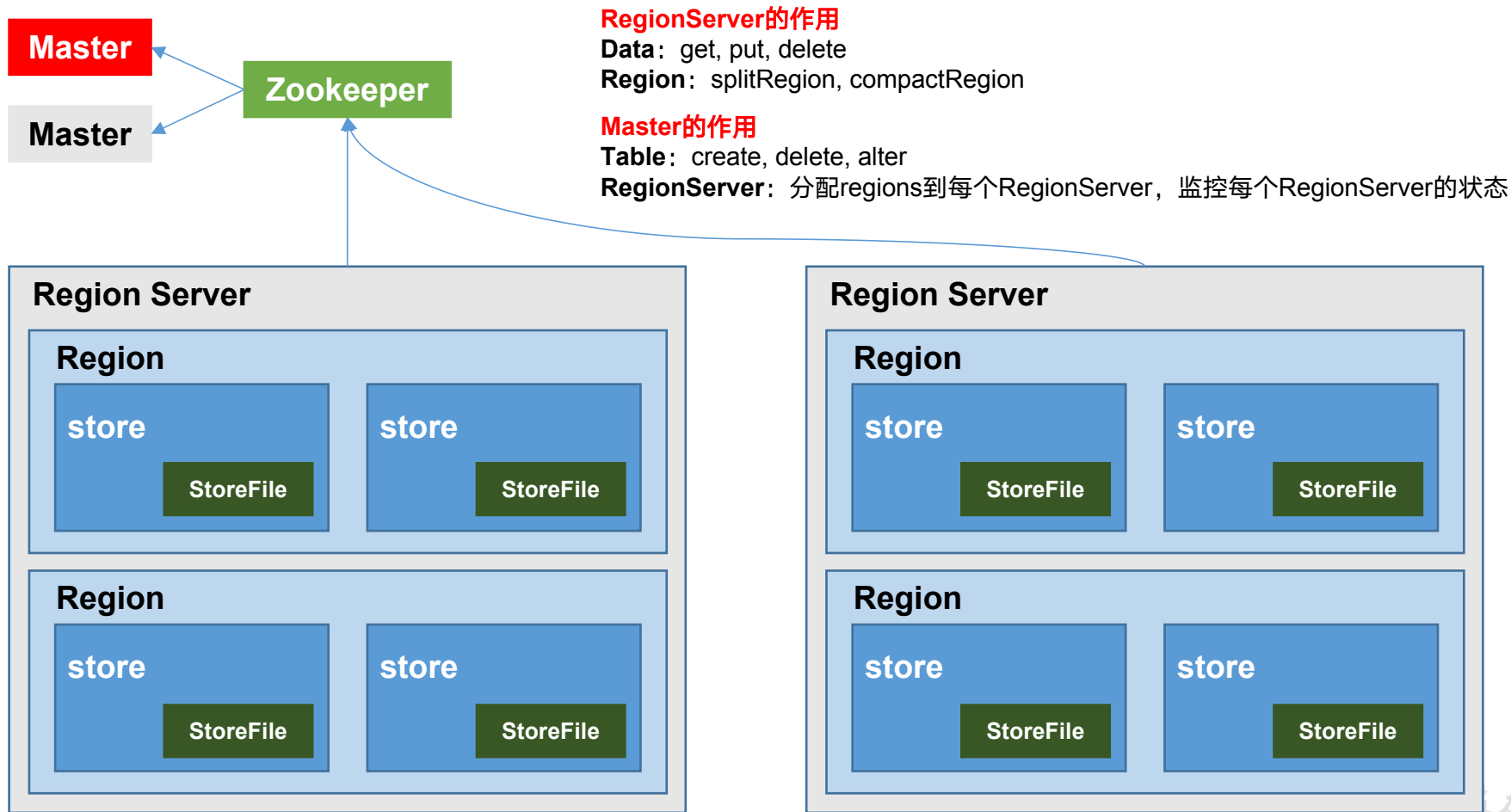
## TimeStamp

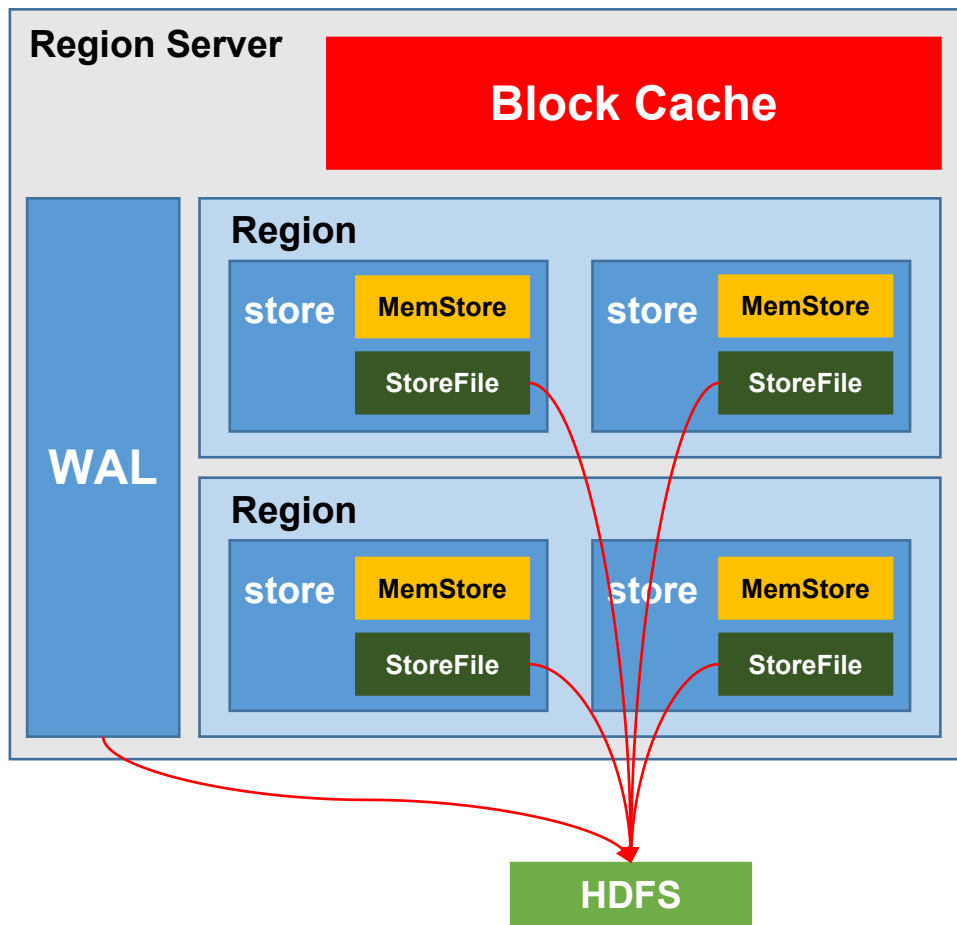
不同版本（version）的数据根据timestamp进行区分

## Type

对于删除操作，其类型为DeleteColumn

Row Key	Column Family	Column Qualifier	TimeStamp	Type	Value
row_key1	personal_info	name	t1	Put	张三
row_key1	personal_info	city	t2	Put	北京
row_key1	personal_info	phone	t3	Put	131*****
row_key1	personal_info	phone	t4	Put	177*****





**StoreFile:** 存储有序的K-V的文件，存储在HDFS上

**MemStore:** 写缓存，K-V在Memstore中进行排序，达到阈值之后才会flush到StoreFile，每次flush生成一个新的StoreFile

**WAL:** Write Ahead Log，预写日志，防止RegionServer故障，导致MemStore中的数据丢失。

**BlockCache:** 读缓存，每次新查询的数据会缓存在BlockCache中。

每个RegionServer可以服务于多个Region

每个RegionServer中有多个Store，  
1个WAL和1个BlockCache

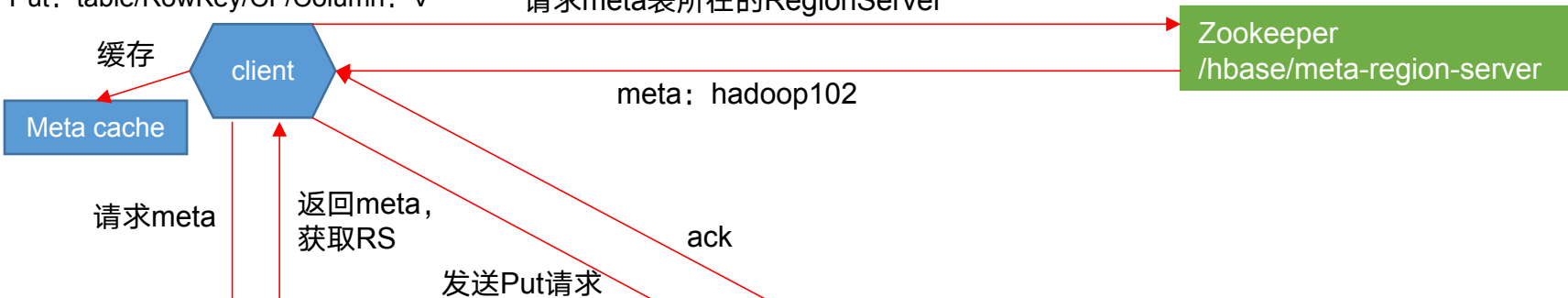
每个Store对应一个列族，包含MemStore和StoreFile

让天下没有难学的技术

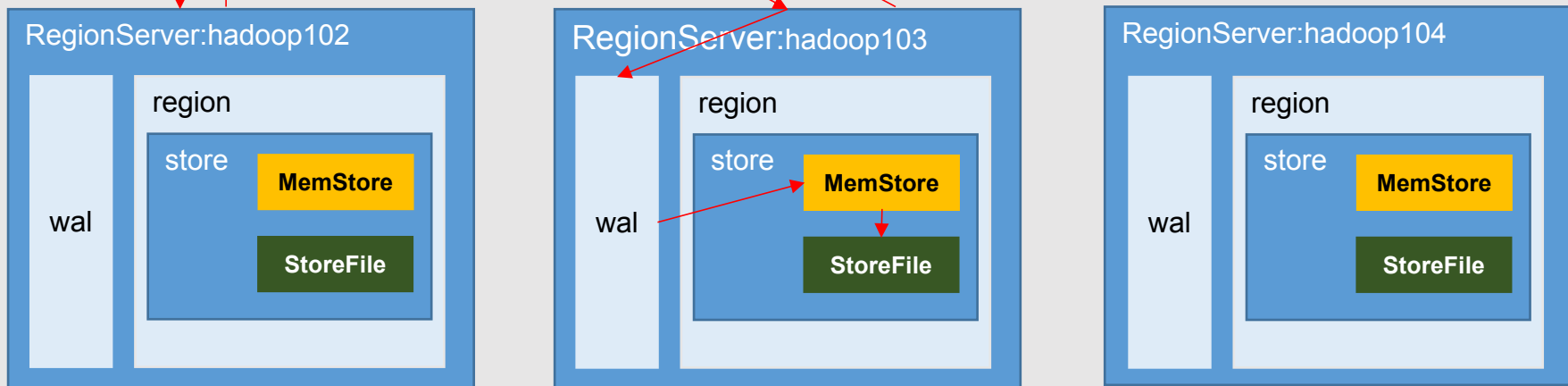


Put: table/RowKey/CF/Column: V

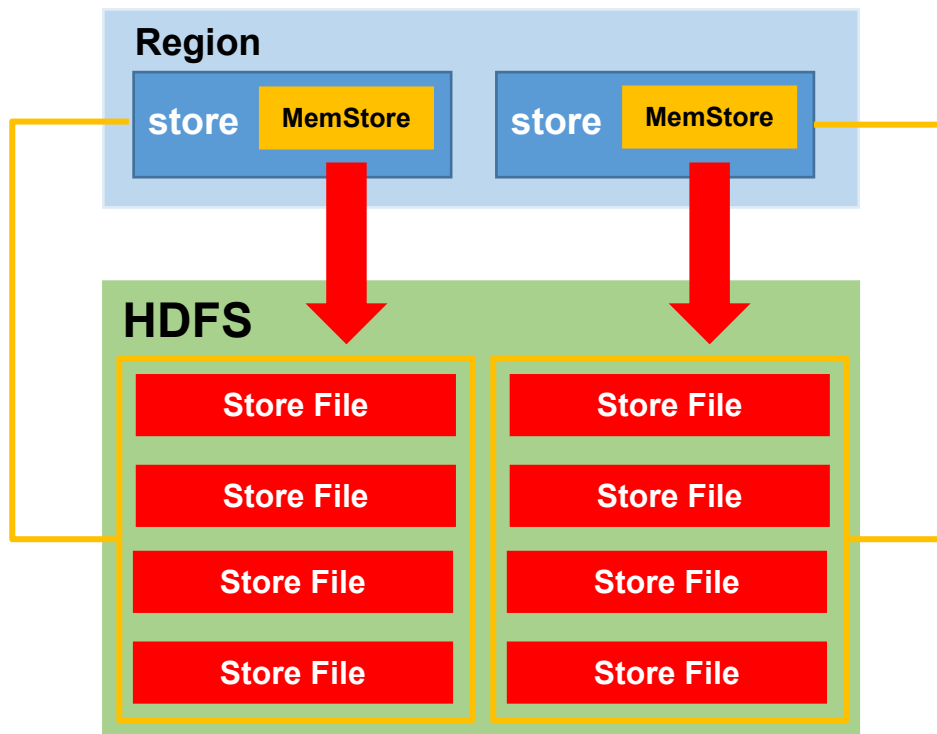
请求meta表所在的RegionServer

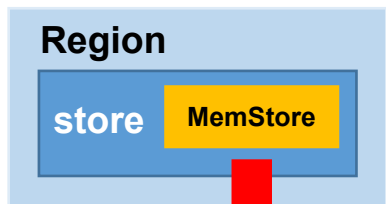


## HBase









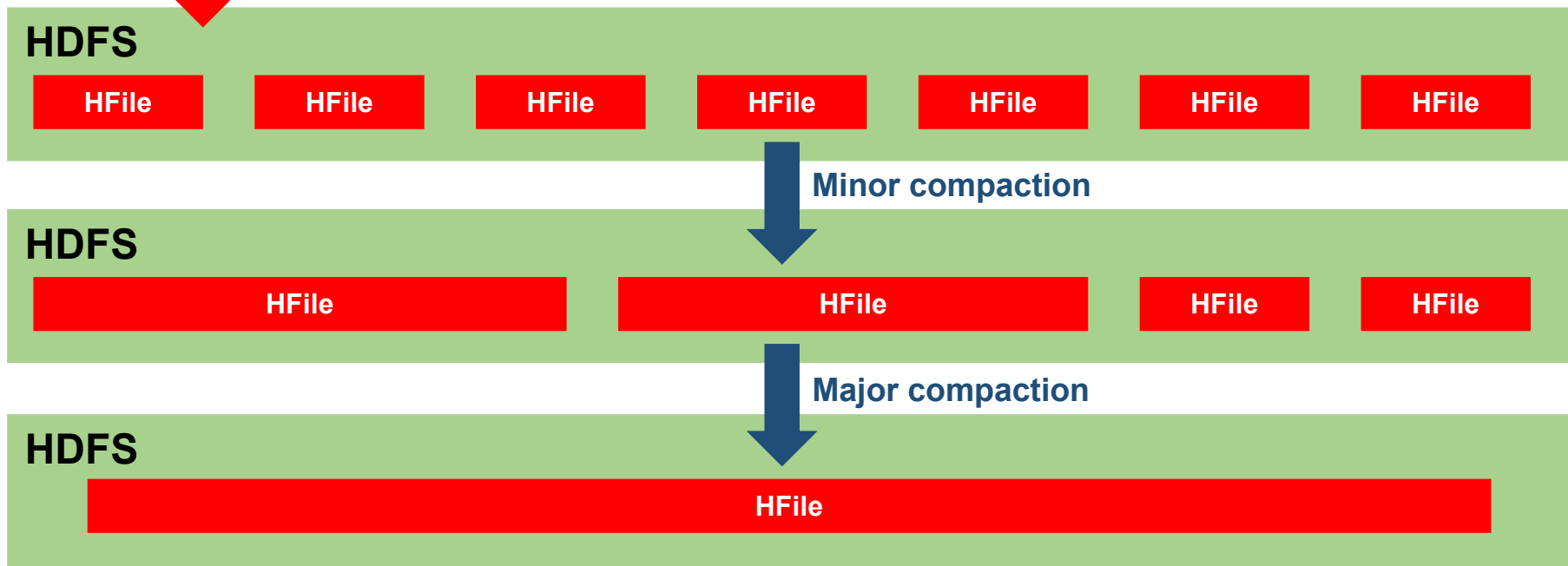
Flush

## Minor compaction

只选取一些小的、相邻的HFile将他们合并成一个更大的HFile，并执行部分的物理删除操作。

## Major compaction

将一个Store下的所有Hfile合并成一个大文件，并执行物理删除操作。





rowkey	info	
	name	age
1	aa	12
2	bb	23
3	cc	34
4	dd	54
5	ee	20
6	ff	12

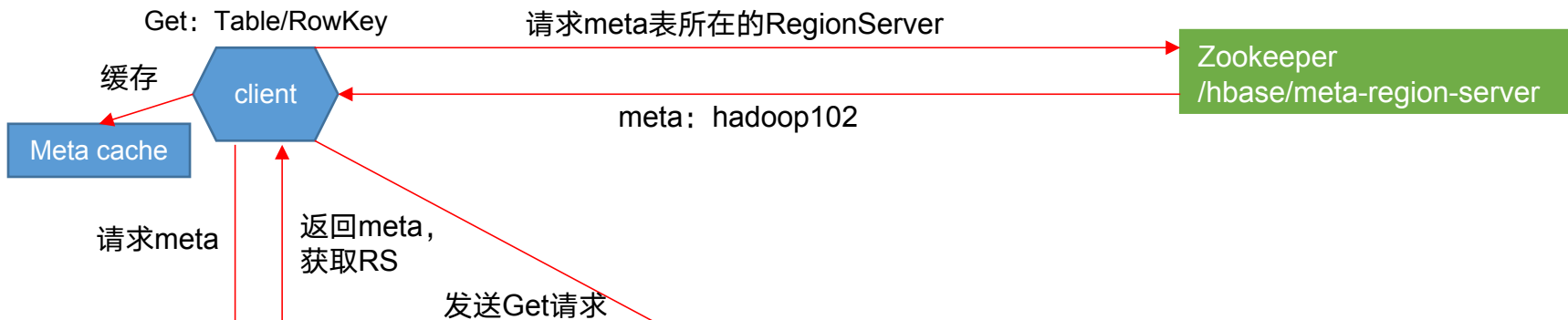
Split

rowkey	info	
	name	age
1	aa	12
2	bb	23
3	cc	34

rowkey	info	
	name	age
4	dd	54
5	ee	20
6	ff	12

1.当1个region中的某个Store下所有StoreFile的总大小超过 “ **hbase.hregion.max.filesize** ” ，该region就会进行拆分（0.94版本之前）。

2.当1个region中的某个Store下所有StoreFile的总大小超过 $\text{Min}(\text{initialSize} \times R^3, \text{hbase.hregion.max.filesize})$ ，该Region就会进行拆分。其中initialSize的默认值为 $2 \times \text{hbase.hregion.max.filesize}$ ，R为当前Region Server中属于该Table的个数（0.94版本之后）。



## HBase

