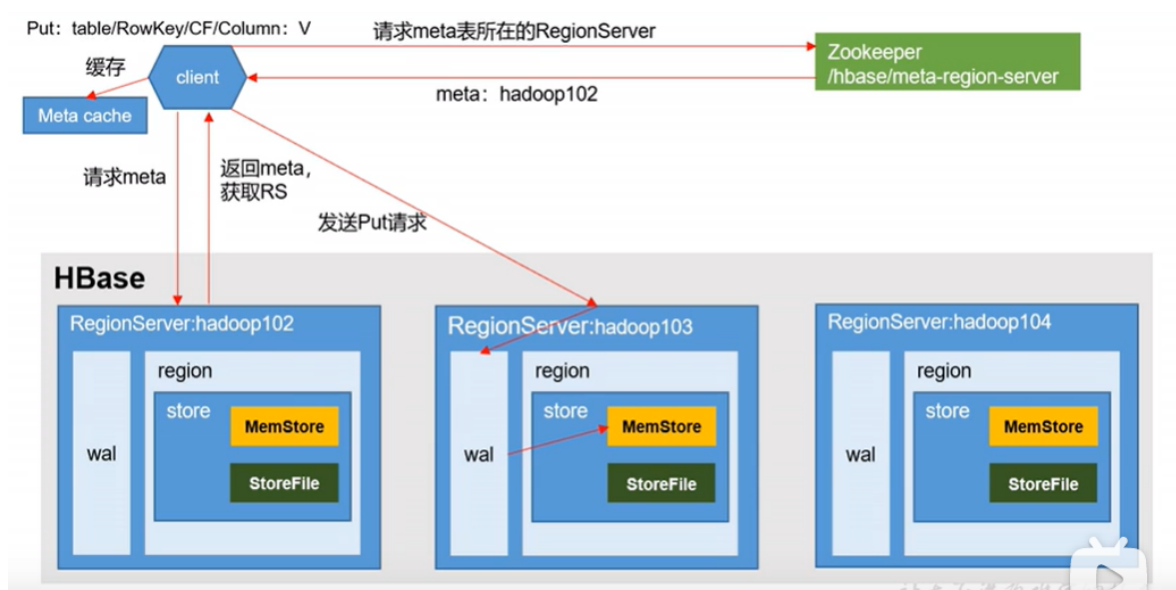


# HBase6 写流程



写入流程框架：

1. client访问zookeeper的meta表所在的Region-Server的位置（meta存放了所在数据所存放的位置，但是因为meta是存储在一个region-server中的，所以需要先找到meta的位置在 /hbase/meta-region-server）
2. zookeeper 将meta存放的region-server的位置告诉client（但是没有meta的内容信息）
3. client通过的到的meta的位置信息去对应的region-server请求meta的内容（根据请求的namespace:table/rowkey，查询出目标数据属于在Region-server的哪个region中，并将信息缓存在客户端的meta cache中）。
4. 对应的region-server将meta的内容（存在需要写入表的位置）返回
5. client通过表的位置信息发送put请求发送给对应的region-server
6. 将数据操作写入Region-server的WAL文件（备份）
7. 将数据写入对应的MemStore，数据会在MemStore进行排序
8. region-server返回ack信息
9. 等待到MemStore的刷写时机后，将数据刷写到HFile

## zookeeper中的/hbase/meta-region-server

输入命令可以得到meta所在的region-server的位置

```
get /hbase/meta-region-server
```

```
[zk: localhost:2181(CONNECTED) 2] get /hbase/meta-region-server
regionserver:16020+un+PBUF

hadoop104}+++.-
cZxid = 0x800000025
ctime = Tue Jul 16 14:12:49 CST 2019
mZxid = 0x800000025
mtime = Tue Jul 16 14:12:49 CST 2019
pZxid = 0x800000025
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 62
numChildren = 0
```

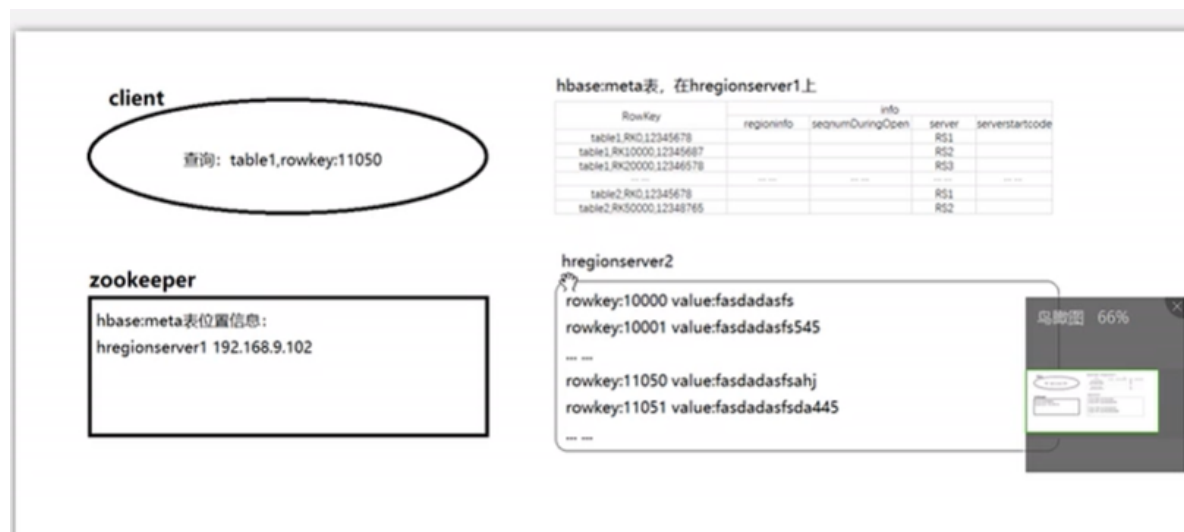
## 查看meta的内容信息

```
scan "hbase:meta"
```

例如我们要修改stu中info的信息，这个时候要看的就是 info: server的值

```
stu,,1563261835567.e9dec column=info:regioninfo, timestamp=1563261836083, value={ENCODED => e9d
f2d4db946585d09b85af36c4 ecf2d4db946585d09b85af36c44e8, NAME => 'stu,,1563261835567.e9decf2d4db
4e8. 946585d09b85af36c44e8.', STARTKEY => '', ENDKEY => ''}
stu,,1563261835567.e9dec column=info:seqnumDuringOpen, timestamp=1563261836083, value=\x00\x00\
f2d4db946585d09b85af36c4 x00\x00\x00\x00\x00\x02
4e8.
stu,,1563261835567.e9dec column=info:server, timestamp=1563261836083, value=hadoop102:16020
f2d4db946585d09b85af36c4
4e8.
stu,,1563261835567.e9dec column=info:serverstartcode, timestamp=1563261836083, value=1563257559
f2d4db946585d09b85af36c4 590
4e8.
```

## 简化流程



查询: table1, rowkey: 10050

1. client去zookeeper找到hbase中的meta表的信息，在hregion-server1中
2. client去hregion-server1上找table1表（因为table1 可能比较大，可能会出现多个分区，这时候根据rowkey查找，找到在hregion-server2）
3. client 去hregion-server2找到要修改的数据

注：可能存在-root- 表，用于存放meta的分区（当表特别大的时候才需要，一般不会出现这么大的表）