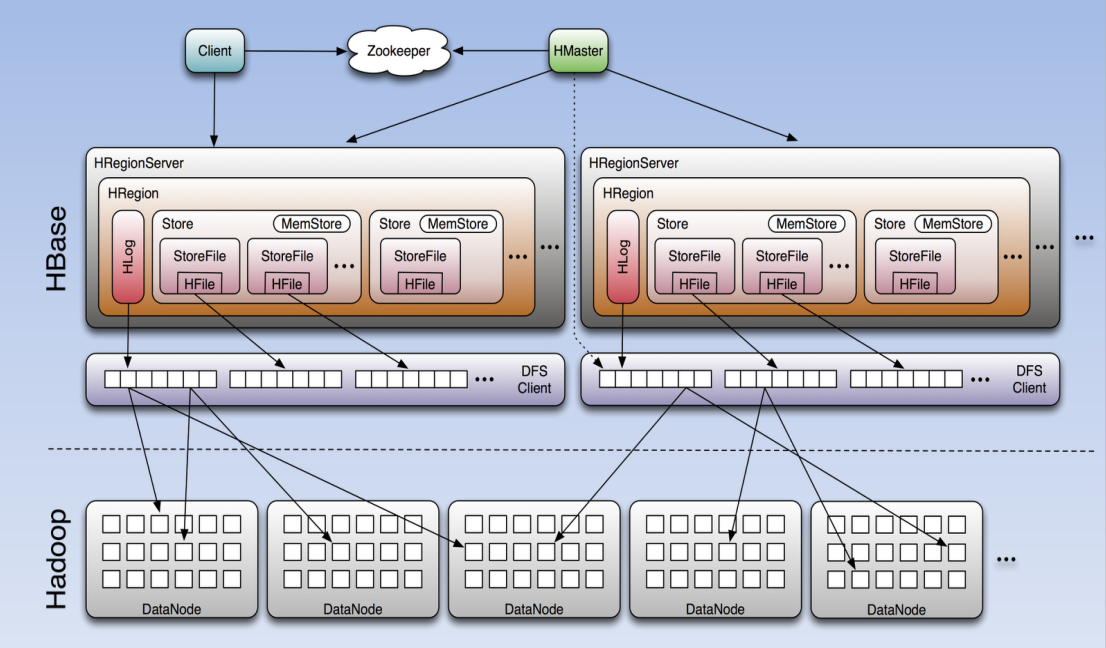
HBase原理： <http://www.blogjava.net/DLevin/archive/2015/08/22/426877.html>

http://www.cnblogs.com/hark0623/p/5571193.html

rowkey设计： http://blog.csdn.net/javajxz008/article/details/51892967

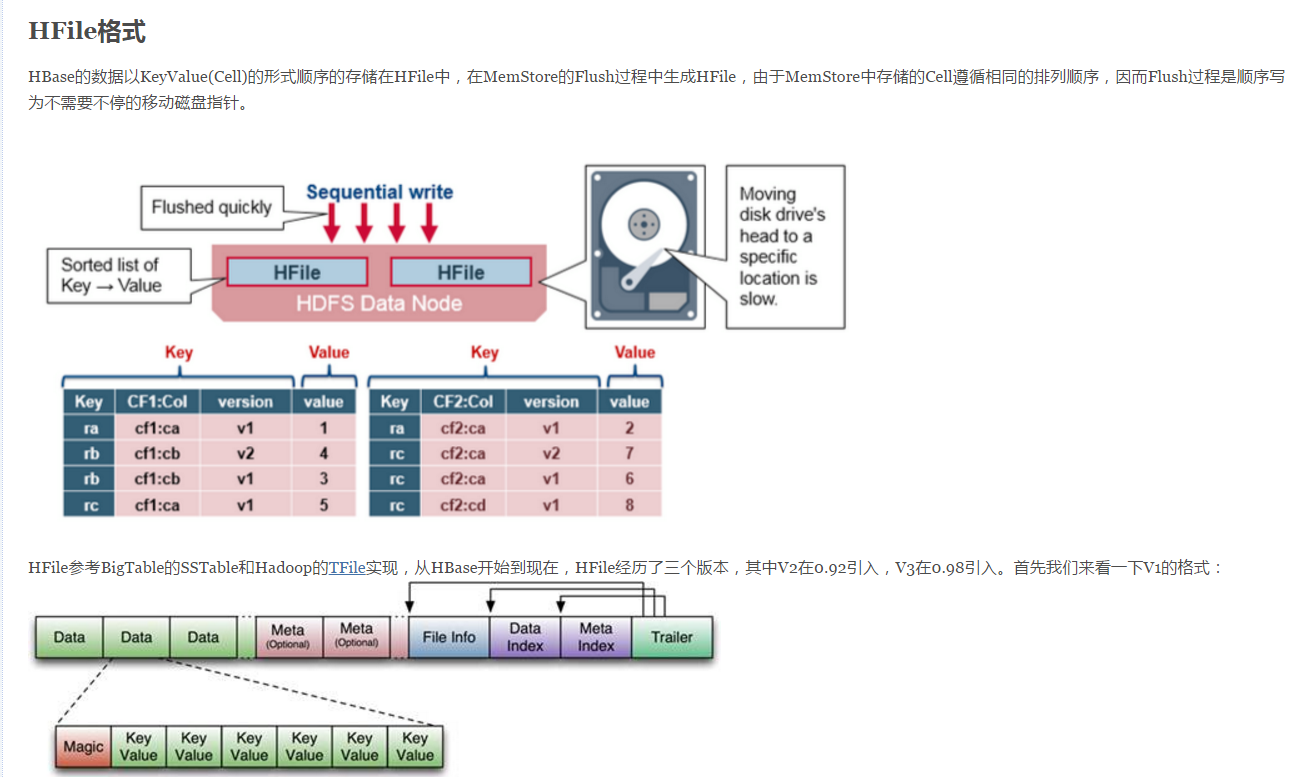
Split参考:https://yq.aliyun.com/articles/43539

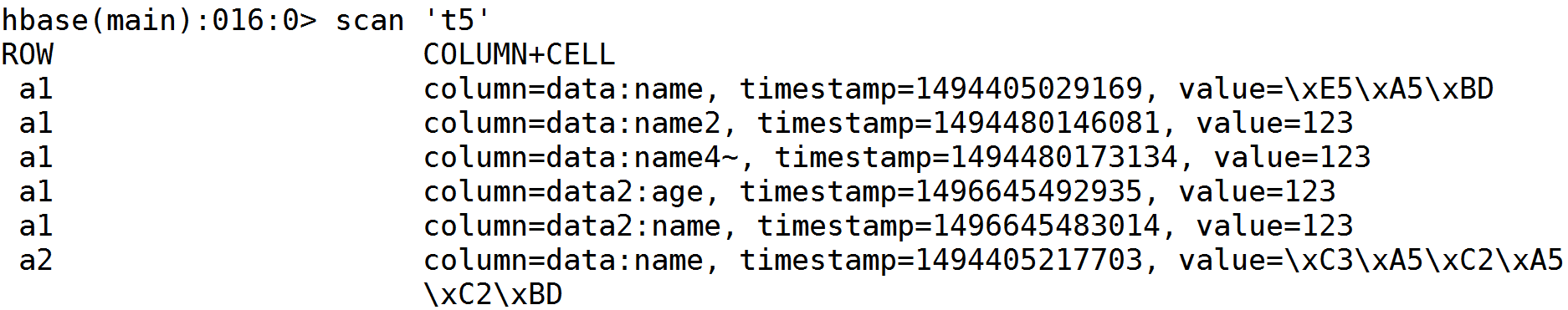


一个表最初由一个HRegion组成，一个HRegion由一个或多个Store，一个Store对应着一个列族，一个Store包含一个或多个StoreFile，StoreFile中是HFile。数据写入时，先经过HMaster找到要写入哪个HRegionServer，再找到哪个HRegion，再找到对应的Store（即写入到哪个列族），将数据写入该Store下的MemStore，进而flush时生成HFile文件，存储到HDFS。读取时类似道理。

HBase的数据最终以KeyValue(Cell)的形式顺序的存储在HFile中，在MemStore的Flush过程中生成HFile。

HFile的存储格式为：Key-Value类型的，如下图所示中的key和value：



Hbase shell，scan结果如下：

Rowkey长度问题：

HBase的数据都是以KeyValue方式存储在MemStore或者HFile中的，每个KeyValue都会存储rowKey的信息，如果rowkey太大的话，比如是128个字节，一行10个字段的表，100万行记录，仅仅rowkey就占了1.2G+，所以长度还是不要过长，按需求来设计。

**HRegion是一个Table中的一个Region在一个HRegionServer中的表达**。一个Table可以有一个或多个Region，他们可以在一个相同的HRegionServer上，也可以分布在不同的HRegionServer上，一个HRegionServer可以有多个HRegion，他们分别属于不同的Table。HRegion由多个Store(HStore)构成，每个HStore对应了一个Table在这个HRegion中的一个Column Family，即每个Column Family就是一个集中的存储单元，因而最好将具有相近IO特性的Column存储在一个Column Family，以实现高效读取(数据局部性原理，可以提高缓存的命中率)。HStore是HBase中存储的核心，它实现了读写HDFS功能，一个HStore由一个MemStore 和0个或多个StoreFile组成。

**MemStore是一个写缓存**(In Memory Sorted Buffer)，所有数据的写在完成WAL日志写后，会 写入MemStore中，由MemStore根据一定的算法将数据Flush到地层HDFS文件中(HFile)，通常每个HRegion中的每个 Column Family有一个自己的MemStore。

1. **HFile(StoreFile) 用于存储HBase的数据(Cell/KeyValue)**。在HFile中的数据是按RowKey、Column Family、Column排序，对相同的Cell(即这三个值都一样)，则按timestamp倒序排列。

