### 1.存储格式

(1)textFile

textFile为默认格式

存储方式：行存储

缺点：磁盘开销大；数据解析开销大；压缩的text文件，hive无法进行合并和拆分

1. Sequencefile

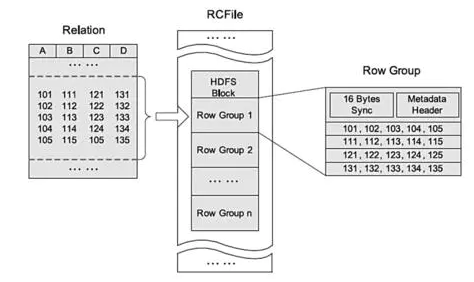
二进制文件，以<key,value>的形式序列化到文件中

存储方式：行存储

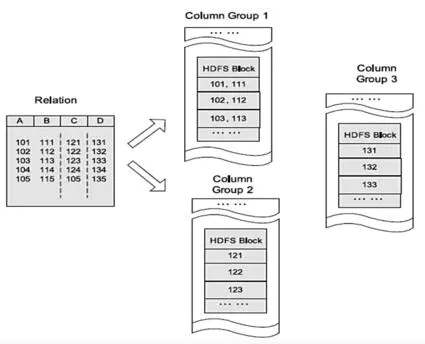
优点：可分割、压缩，一般选择block压缩，与hadoop api中的mapfile是互相兼容的。

#### Rcfile

一种行列存储相结合的存储方式。首先，其将数据按行分块。其次，块数据列式存储，有利于数据压缩和快速的列存取。



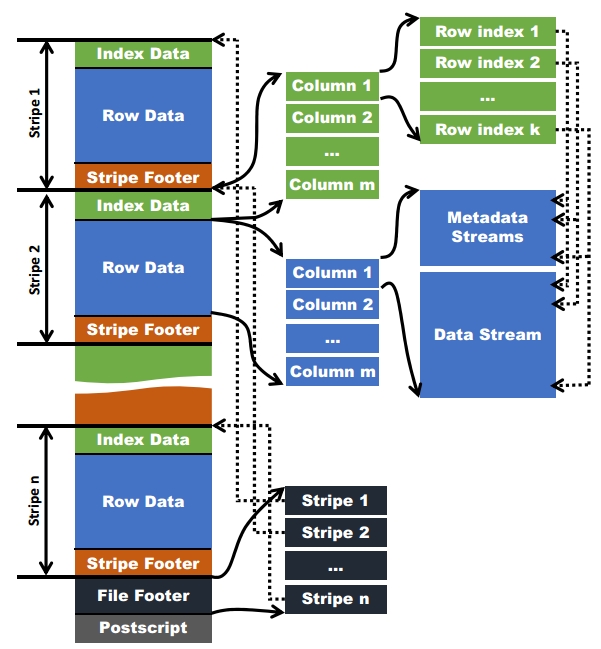
HDFS Blocks是16字节的HDFS同步块信息，主要包括该行组内的存储的行数、列的字段信息等等。优势 在一般的行存储中 select a from table，虽然只是取出一个字段的值，但是还是会遍历整个表，所以效果和select \* from table 一样，在RCFile中，像前面说的情况，只会读取该行组的一行。



**劣势 ：**拷贝到本地目录后RCFile并不是真正直接跳过不需要的列，并跳到需要读取的列， 而是通过扫描每一个row group的头部定义来实现的，所以在读取所有列的情况下，RCFile的性能反而没有SequenceFile高。

#### (4)ORC

hive给出的新格式，属于RCFILE的升级版



Postscripts中存储该表的行数，压缩参数，压缩大小，列等信息

Stripe Footer中包含该stripe的统计结果，包括Max，Min，count等信息

FileFooter中包含该表的统计结果，以及各个Stripe的位置信息

IndexData中保存了该stripe上数据的位置信息，总行数等信息

RowData以stream的形式保存了数据的具体信息

**ORC文件的元数据了解了一个ORC文件会被分成多个stripe，而且文件的元数据中有每个字段的统计信息（min/max,hasNull等等），这就为ORC的查询优化做好了基础准备。假如我的查询过滤条件为WHERE id = 0;在Map Task读到一个ORC文件时，首先从文件的统计信息中看看id字段的min/max值，如果0不包含在内，那么这个文件就可以直接跳过了。**

数据访问

　　读取ORC文件是从尾部开始的，第一次读取16KB的大小，尽可能的将Postscript和Footer数据都读入内存。文件的最后一个字节保存着PostScript的长度，它的长度不会超过256字节，PostScript中保存着整个文件的元数据信息，它包括文件的压缩格式、文件内部每一个压缩块的最大长度(每次分配内存的大小)、Footer长度，以及一些版本信息。在Postscript和Footer之间存储着整个文件的统计信息(上图中未画出)，这部分的统计信息包括每一个stripe中每一列的信息，主要统计成员数、最大值、最小值、是否有空值等。

　　接下来读取文件的Footer信息，它包含了每一个stripe的长度和偏移量，该文件的schema信息(将schema树按照schema中的编号保存在数组中)、整个文件的统计信息以及每一个row group的行数。

### 2 创建orc格式表

hive> create table if not exists t2(id bigint, name string, age int)

> row format delimited fields terminated by ','

> stored as orc;

**用ORC格式，需先将数据导入textfile格式的临时表**

hive> insert into t2 select \* from t1;

查看数据

hive> select \* from t2;

OK

10001 zs 20

20002 ls 30

30003 ww 40

40004 zl 50

50005 tq 60

60006 qq 70

查看表详细信息

hive> desc formatted t2;