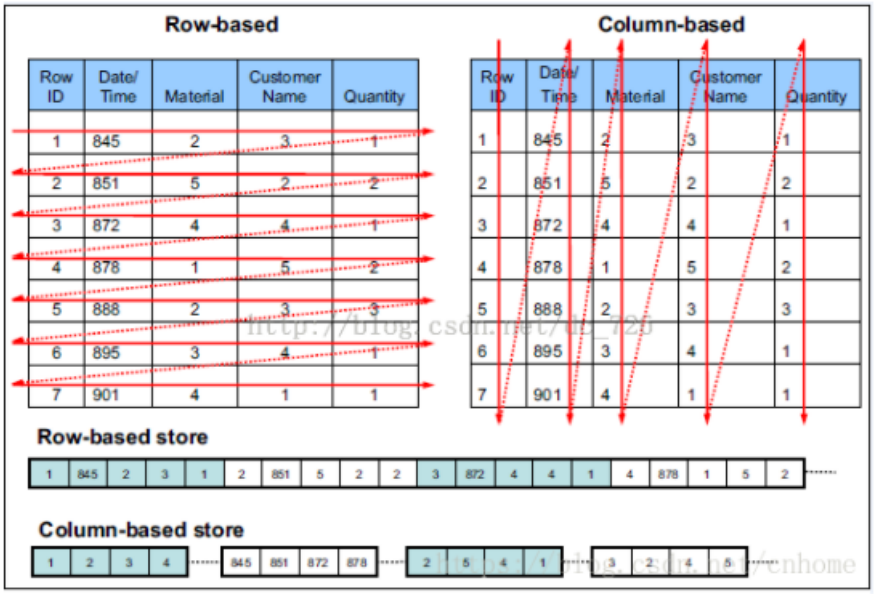
Hive总结5 数据存储格式

> Hive支持的存储数据的格式主要有：TEXTFILE（行式存储） 、SEQUENCEFILE(行式存储)、ORC（列式存储）、PARQUET（列式存储）。

列式存储和行式存储



行储存的特点

查询满足条件的一整行数据的时候，行存储只需要找到其中一个值，其余的值都在相邻地方。列存储则需要去每个聚集的字段找到对应的每个列的值，所以此时行存储查询的速度更快。

列储存的特点

因为每个字段的数据聚集存储，在查询只需要少数几个字段的时候，能大大减少读取的数据量；每个字段的数据类型一定是相同的，列式存储可以针对性的设计更好的设计压缩算法。

TEXTFILE和SEQUENCEFILE的存储格式都是基于行存储的；

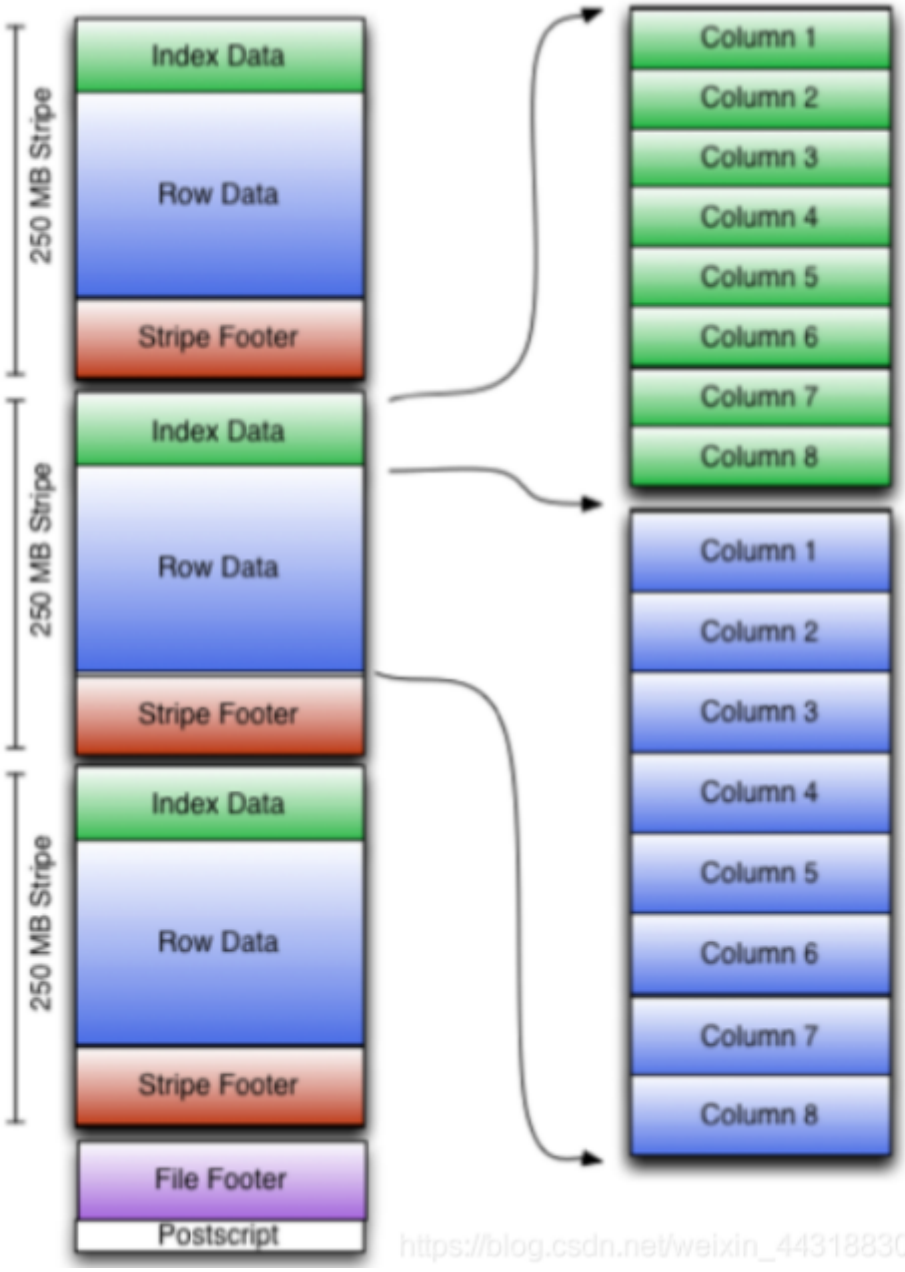
ORC和PARQUET是基于列式存储的。

TEXTFILE格式

默认格式，数据不做压缩，磁盘开销大，数据解析开销大。可结合Gzip、Bzip2使用(系统自动检查，执行查询时自动解压)，但使用这种方式，hive不会对数据进行切分，从而无法对数据进行并行操作。

ORC格式

可以看到每个Orc文件由1个或多个stripe组成，每个stripe250MB大小，这个Stripe实际相当于RowGroup概念，不过大小由4MB->250MB，这样能提升顺序读的吞吐率。每个Stripe里有三部分组成，分别是Index Data,Row Data,Stripe，Footer：



主要部分：

- 一个orc文件可以分为若干个Stripe
- 一个stripe可以分为三个部分
    - **Index Data**：一个轻量级的index，默认是每隔1W行做一个索引。这里做的索引只是记录某行的各字段在Row Data中的offset。
    - **Row Data**：存的是具体的数据，先取部分行，然后对这些行按列进行存储。对每个列进行了编码，分成多个Stream来存储。
    - **Stripe Footer**：存的是各个stripe的元数据信息

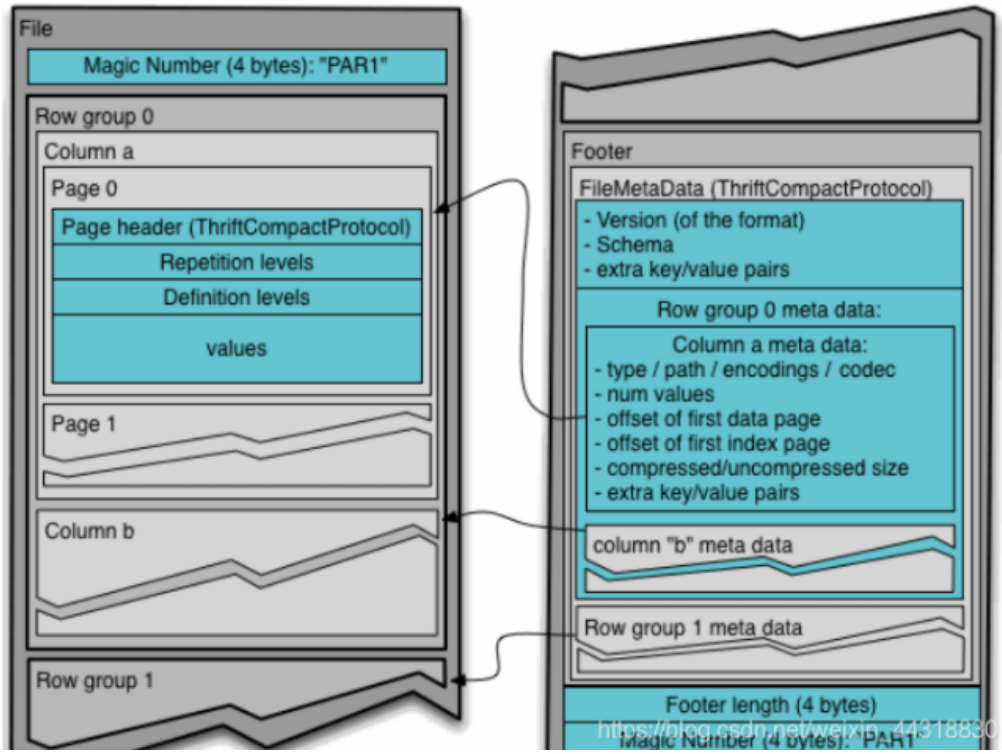
每个文件有一个File Footer，这里面存的是每个Stripe的行数，每个Column的数据类型信息等；

每个文件的尾部是一个PostScript，这里面记录了整个文件的压缩类型以及FileFooter的长度信息等。在读取文件时，会seek到文件尾部读PostScript，从里面解析到File Footer长度，再读FileFooter，从里面解析到各个Stripe信息，再读各个Stripe，即从后往前读。

PARQUET格式

Parquet是面向分析型业务的列式存储格式。Parquet文件是以二进制方式存储的，所以是不可以直接读取的，文件中包括该文件的数据和元数据，因此Parquet格式文件是自解析的。

通常情况下，在存储Parquet数据的时候会按照Block大小设置行组的大小，由于一般情况下每一个Mapper任务处理数据的最小单位是一个Block，这样可以把每一个行组由一个Mapper任务处理，增大任务执行并行度。Parquet文件的格式



Parquet文件的内容，一个文件中可以存储多个行组，文件的首位都是该文件的Magic Code，用于校验它是否是一个Parquet文件，Footer length记录了文件元数据的大小，通过该值和文件长度可以计算出元数据的偏移量，文件的元数据中包括每一个行组的元数据信息和该文件存储数据的Schema信息。除了文件中每一个行组的元数据，每一页的开始都会存储该页的元数据，在Parquet中，有三种类型的页：数据页、字典页和索引页。数据页用于存储当前行组中该列的值，字典页存储该列值的编码字典，每一个列块中最多包含一个字典页，索引页用来存储当前行组下该列的索引，目前Parquet中还不支持索引页。

存储文件的压缩比测试：

> 一个原始数据为19M的数据为例

1. 创建表，存储数据格式为TEXTFILE

```
create table log_text2 (
track_time string,
url string,
session_id string,
referer string,
ip string,
end_user_id string,
city_id string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE ;
```

```
create table log_orc(
track_time string,
url string,
session_id string,
referer string,
ip string,
end_user_id string,
city_id string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS orc ;
```

```
create table log_parquet(
track_time string,
url string,
session_id string,
referer string,
ip string,
end_user_id string,
city_id string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS PARQUET
```

2. 向表中加载数据

```
load data local inpath '/export/servers/hivedatas/log.data' into table
log_text1 ;
```

```
insert into table log_orc select * from log_text1 ;
```

```
insert into table log_parquet select * from log_text ;
```

3. 查看表数据的大小

```
dfs -du -h /user/hive/warehouse/myhive.db/log_text;
```

```
dfs -du -h /user/hive/warehouse/myhive.db/log_orc;
```

```
dfs -du -h /user/hive/warehouse/myhive.db/log_parquet;
```

textfile压缩之后是18.1M

orc压缩后是2.8M

parquet压缩后是13.1M

存储文件的查询速度测试

```
hive (default)> select count(*) from log_text;

结果:
_c0
100000

1 row selected (5.97 seconds)
1 row selected (5.754 seconds)
```

```
hive (default)> select count(*) from log_orc;


结果:
_c0
100000

1 row selected (5.967 seconds)
1 row selected (6.761 seconds)
```

```
hive (default)> select count(*) from log_parquet;

结果:
_c0
100000

1 row selected (6.7 seconds)
1 row selected (6.812 seconds)
```

总结

```
压缩效率比较:
ORCR > parquet > textFile
查询速度比较:
TextFile >ORC> Parquet
```