Apache Cassandra SSTable 存储格式详解

在 Cassandra 中，当达到一定条件触发 flush 的时候，表对应的 Memtable 中的数据会被写入到这张表对应的数据目录（通过 data_file_directories 参数配置）中，并生成一个新的 SSTable（Sorted Strings Table，这个概念是从 Google 的 BigTable 借用的）。每个 SSTable 是由一系列的不可修改的文件组成，这些文件在 Cassandra 中被称为 Component。本文是基于 Cassandra 3.11.4 版本介绍的，这个版本生成的 SSTable 由以下 Component 组成：

```
-rw-r--r-- 1 iteblog iteblog   43 May  3 11:55 md-1-big-CompressionInfo.db
-rw-r--r-- 1 iteblog iteblog  155 May  3 11:55 md-1-big-Data.db
-rw-r--r-- 1 iteblog iteblog   10 May  3 11:55 md-1-big-Digest.crc32
-rw-r--r-- 1 iteblog iteblog   16 May  3 11:55 md-1-big-Filter.db
-rw-r--r-- 1 iteblog iteblog   66 May  3 11:55 md-1-big-Index.db
-rw-r--r-- 1 iteblog iteblog 4866 May  3 11:55 md-1-big-Statistics.db
-rw-r--r-- 1 iteblog iteblog   98 May  3 11:55 md-1-big-Summary.db
-rw-r--r-- 1 iteblog iteblog   92 May  3 11:55 md-1-big-TOC.txt
```

从上面文件名可以看出，所有的 Component 都是由 md-1-big 开头的，每个文件名都是由 version - generation - SSTable Format type - Component name 组成，由 - 分割，每个字符的含义如下：

- version：这个代表 SSTable 存储格式的版本号，目前最新的 Cassandra 3.11.4 版本为 md，其他有效的版本为 jb、ka、la、lb、ma、mb 以及 mc 等，比如 Cassandra 3.0.8 的版本就是 mc。具体参见 org.apache.cassandra.io.sstable.format.big.BigFormat.BigVersion；
- generation：代表索引号，每次生成新的 SSTable 时，这个索引号都会递增；
- SSTable Format type：SSTable格式类型，主要有两种 BIG 和 LEGACY，但是他们的名字都是 big；
- Component name：代表当前文件存储的信息类型，比如 Data 代表存储的是用户实际写入的数据；Index 代表存储的是索引数据等。当前版本的 Cassandra 组件有 Data.db、Digest.crc32、CompressionInfo.db、Summary.db、Filter.db、Index.db、TOC.txt 以及 Statistics.db。

上面这些组件共同构建 SSTable，每种文件存储不同类型的数据，本文就是来介绍这些文件的作用以及文件之间的关联。

md-1-big-Data.db、md-1-big-Index.db 以及 md-1-big-Summary.db 文件介绍

md-1-big-Data.db

从名字就可以看出，md-1-big-Data.db 是存储用户插入到 Cassandra 对应表中数据的，这个文件的数据存储格式我们在 《Apache Cassandra 数据存储模型》 这篇文章中已经详细介绍了。

为了有个更直观的了解，假设我们有一张名为 iteblog_test 的表，建表语句和导入数据的语句如下：

```
create table iteblog_test(username text, type text, email text, age text, cril text, briday text ,PRIMARY KEY(username, type));

insert into iteblog_test(username, type, email) values ('iteblog', '456', 'wyphao.2007@163.com');
insert into iteblog_test(username, type, email, age) values ('iteblog', '123', 'hadoop@spark.org', '99');
insert into iteblog_test(username, type, briday) values ('cassandra', '246', '2019-04-29');
```

我们对这张表执行 flush 操作，这时候底层生成的 md-1-big-Data.db 内容如下：

```
00000000  87 00 00 00 f2 00 00 07  69 74 65 62 6c 6f 67 7f  |........iteblog.|
00000010  ff ff ff 80 00 01 00 fb  3d 04 00 03 31 32 33 1a  |........=...123.|
00000020  15 90 86 02 08 02 39 39  08 10 68 61 64 6f 6f 70  |......99..hadoop|
00000030  40 73 70 61 72 6b 2e 6f  72 67 04 00 03 34 35 36  |@spark.org...456|
00000040  18 21 00 03 08 13 77 79  70 68 61 6f 2e 32 30 30  |.!....wyphao.200|
00000050  37 40 31 36 33 2e 63 6f  6d 01 00 09 63 61 73 73  |7@163.com...cass|
00000060  61 6e 64 72 61 58 00 f0  08 32 34 36 12 17 e0 24  |andraX...246...$|
00000070  4d 75 05 08 0a 32 30 31  39 2d 30 34 2d 32 39 01  |Mu...2019-04-29.|
00000080  ac b6 4c 9c                                       |..L.|
00000084
```

注意：上面内容分为三部分，最左边那列（00000000 - 00000084）这个代表的是十六进制的行号；中间两列（87 00 00 00 f2 00 00 07 69 74 65 62 6c 6f 67 7f）这种是代表 md-1-big-Data.db 文件的十六进制表示；最后那列（两个 | 之间的数据）代表中间两列十六进制对应的 ASCII 字符。

从最后边那列可以看出，md-1-big-Data.db 文件存储了用户插入到 iteblog_test 表的数据，并且同一个 Partition Key 以及静态列（如果有）只会在同一个 SSTable 中存储一份；Partition Key 相同的行对应的 Clustering key 是有序排序的。比如上面 Partition Key 为 iteblog 对应了两行数据，他们在 md-1-big-Data.db

文件是按照字符串的字典顺序升序排序的。关于 md-1-big-Data.db 文件的底层存储格式可以参见《Apache Cassandra 数据存储模型》的说明，这里就不再详细介绍了。

md-1-big-Index.db

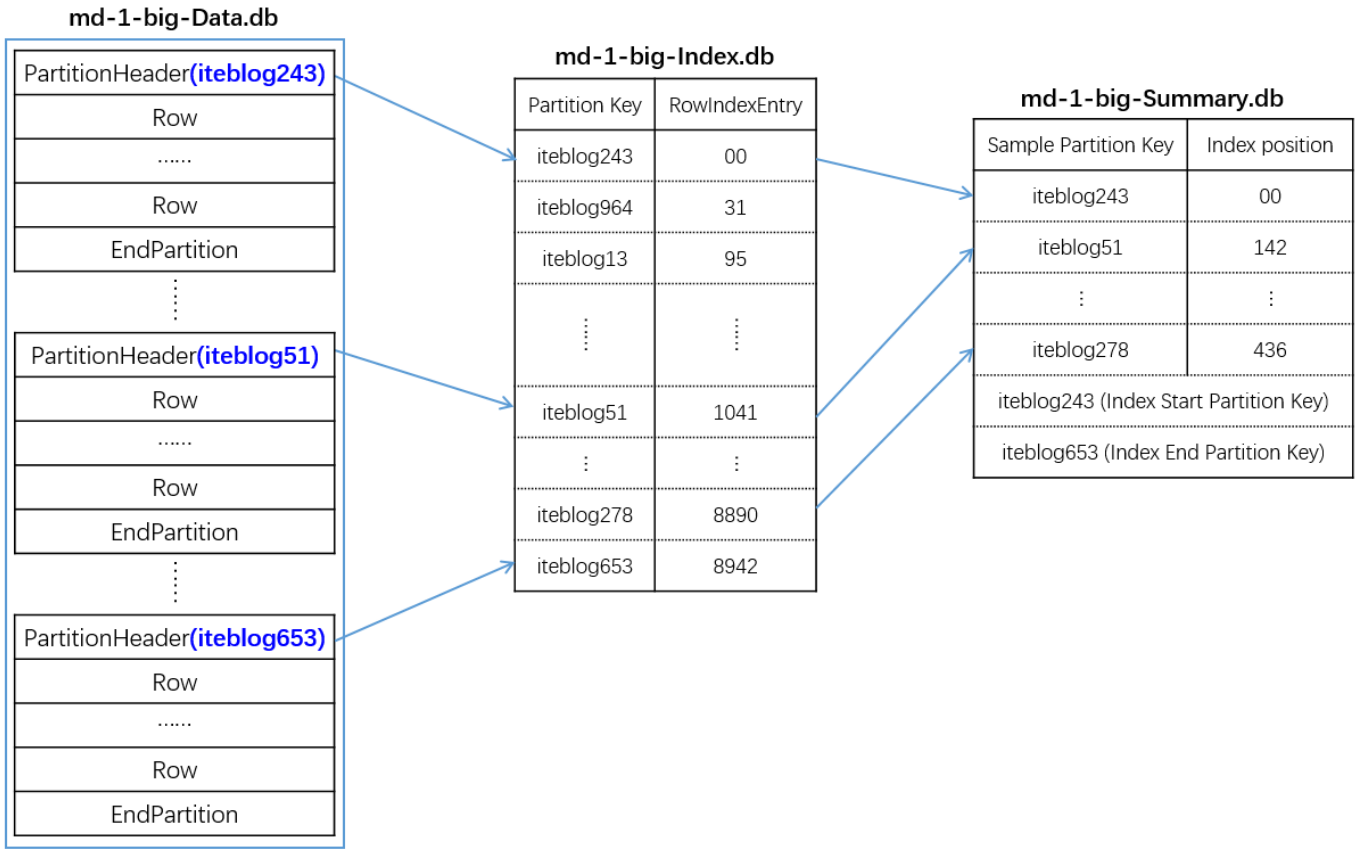
SSTable 对应的 md-1-big-Data.db 可能会很大，为了加快查询速度，Cassandra 对 md-1-big-Data.db 文件中的每个 Partition Key 建立了相关索引数据，这个就是 md-1-big-Index.db 文件的作用了。md-1-big-Index.db 文件存储的内容其实很简单，存储的是 Partition Key 及其在 md-1-big-Data.db 文件中的起始偏移量。

md-1-big-Summary.db

如果 md-1-big-Index.db 文件也很大，也可能会影响查询速度。所以 Cassandra 引入了 md-1-big-Summary.db 文件，对索引文件 md-1-big-Index.db 进行抽样。具体过程为：每隔 128（这个数由 DEFAULT_MIN_INDEX_INTERVAL 决定，好像不可以配置）个 Partition Key 就将对应的 Partition Key 及其在索引文件中的起始偏移量写入到 Summary.db 文件中去。同时，Summary.db 文件最后面还会存储 md-1-big-Index.db 文件中的起止 Partition Key。

这三个文件的关系

前面已经简单介绍了这三个文件的作用，为了直观的表示这是三个文件的关系，我这里画了一张图，帮助大家理解。



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公众号：iteblog_hadoop

注意：上面的 iteblogxxx 代表的是 Partition Key，并且假设这些 Partition Key 对应的 hash token 是升序的。

md-1-big-Statistics.db

包含有关 SSTable 的统计元数据的文件。主要包含当前 SSTable 的：

- 最大最小 token 的值及对应的 Partition Key；
- Partitioner，默认为 org.apache.cassandra.dht.Murmur3Partitioner，和 Bloom Filter FP chance，主要用于校验 SSTable，在读取 Data.db 之前会先被读取；
- SSTable 元数据，比如 SSTable 最大最小的时间戳、最大最小 local deletion time、最大最小的 TTL、SSTable 文件的压缩率、行数、列个数以及最大和最小的 ClustringValues，如果为空这保存为 0 等等
- Partition Key 的类型；
- Clustering Key 的类型；
- 静态列的名字及类型；
- 正常类的名字及类型；
- Clustering Key 的类型；

md-1-big-CompressionInfo.db

存储 SSTable
的压缩相关的信息，包括使用的压缩算法的名字（LZ4Compressor，SnappyCompressor 和 DeflateCompressor，SSTable 的压缩算法可以在创建表的时候通过 WITH compression = {'sstable_compression': 'SnappyCompressor' 设置，默认为 LZ4Compressor），压缩算法的配置选项，chunkLength（默认为 65536），未压缩数据的长度，chunk 的个数等相关的信息

md-1-big-Filter.db、md-1-big-Digest.crc32

md-1-big-Filter.db

存储 Partition Key
布隆过滤器相关的信息，这个文件会被加载到内存，在查询的时候可以确定某行数据是否在这个 SSTable 中，减少访问磁盘的次数。

md-1-big-Digest.crc32

这个文件存储的仅仅是数据文件的 CRC 校验码信息。

md-1-big-TOC.txt

这个文件主要存储 SSTable 的 Component 名字，所以我们打开这个文件可以看到如下的内容：

Statistics.db
Digest.crc32
Data.db
Filter.db
CompressionInfo.db
Summary.db
TOC.txt
Index.db

这个就是 SSTable 对应的八个 Component 。