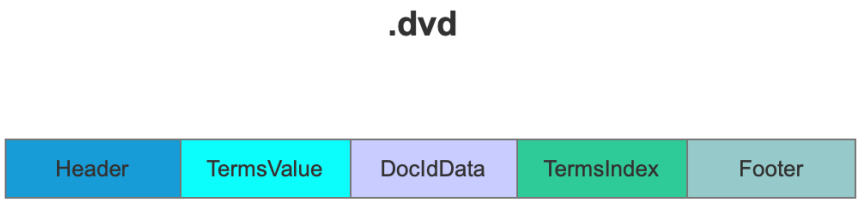
BinaryDocValues

BinaryDocValues同 SortedDocValues、SortedNumericDocValues一样，在实际应用中最多的场景用于提供给搜索结果一个排序规则。在搜索结果的排序阶段，实际是按照BinaryDocValuesFiled的域值根据字典序进行比较，而SortedDocValues则是比较域值对应的ord值来作比较，所以BinaryDocValues在排序上性能远不如SortedDocValues，当然BinaryDocValues最突出的优点对固定长度域值的索引最大化的优化，看完下面的介绍后就明白了。在后面的文章中会详细介绍BinaryDocValues的应用，在这里重点是讲述其生成的索引文件数据结构。

数据结构

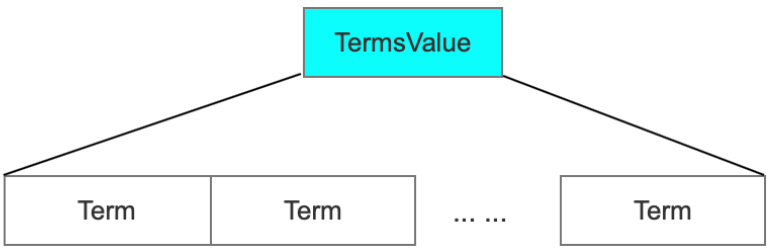
dvd

图1：



TermsValue

BinaryDocValuesFiled同SortedDocValuesFiled一样都是单值域，即一篇文档中只能有一个相同域名的域值，所以TermsValue中记录的就是每篇文档中的完整域值，并且不使用前缀存储，这里强调的是，TermsValue只记录域值，即所有的域值都是挨个写入到.dvd文件中，不同其他索引文件，一般索引文件都会先记录一个域值的长度，然后再记录域值的值，这样在读取阶段才能明确在.dvd文件中读取的数据区间。BinaryDocValues则是通过TermsIndex字段来映射，随后会介绍。 图2：



Term

每个BinaryDocValuesFiled的完整域值，并且可能有重复的域值。

DocIdData

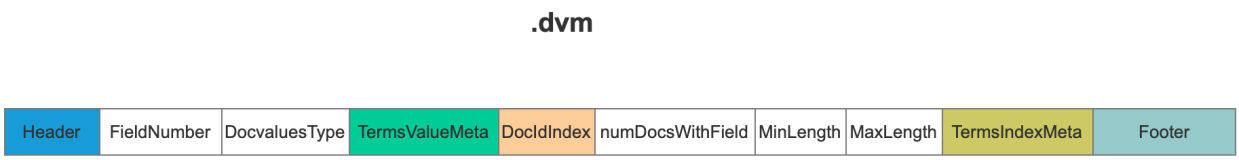
DocIdData中记录包含当前域的文档号。 如果IndexWriter添加的document中不都包含当前域，那么需要将包含当前域的文档号记录到DocIdData中，并且使用IndexedDISI类来存储文档号，IndexedDISI存储文档号后生成的数据结构单独的作为一篇文章介绍，在这里不赘述。

TermsIndex

TermsIndex中记录了TermsValue中每一个域值相对于第一个域值的在.dvd文件中的偏移，在读取阶段，如果需要读取第n个域值的数据，那么只要计算 第n个和第n+1个的TermsIndex的差值，就可以获得第n个域值在.dvd文件中的数据区间，实现随机访问。源码中实际是一个下标值为文档号，数组元素是TermsIndex的数组结构，每一个域值的TermsIndex采用PackedInts进行压缩存储。其数据结构在其他DocValues的文章中有介绍，这里不赘述。

dvm

图3：



FieldNumber

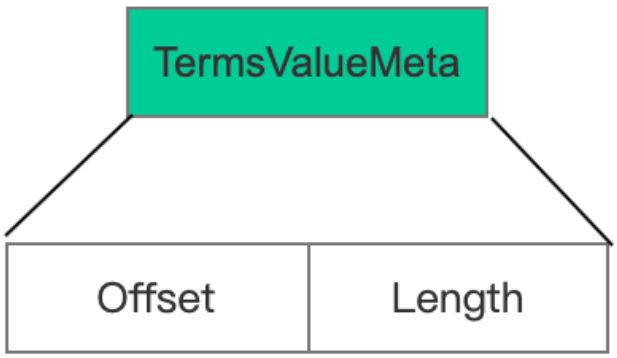
域的编号

DocvaluesType

Docvalues的类型，本文中，这个值就是 BINARY。

TermsValueMeta

图4：



offset

```
.dvd文件中TermsValue在文件中的开始位置。
```

length

```
length为 TermsValue在.dvd文件中的数据长度。
```

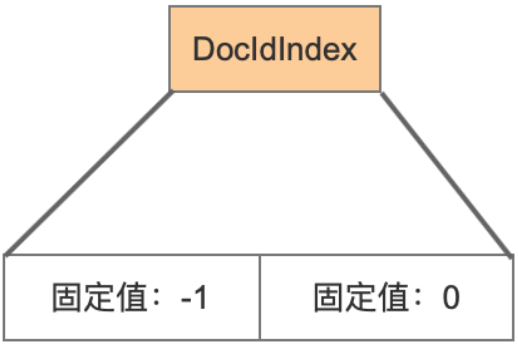
在读取阶段，通过offset跟length就可以获得所有的 TermsValue数据。

DocIdIndex

DocIdIndex是对.dvd文件的一个索引，用来描述 .dvd文件中DocIdData在.dvd文件中的开始跟结束位置。

情况1：

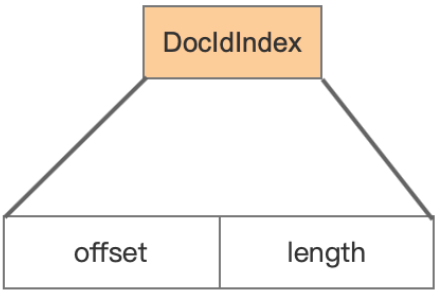
图5：



如果IndexWriter添加的document中都包含当前域，那么只需要在DocIdIndex中添加标志信息即可。

情况2：

图6：



如果IndexWriter添加的document中不都包含当前域，那么.dvd文件中需要将包含当前的域的文档号信息都记录下来。

offset

```
.dvd文件中存放文档号的DocIdData在文件中的开始位置。
```

length

```
length为DocIdData在.dvd文件中的数据长度。
```

在读取阶段，通过offset跟length就可以获得所有的DocIdData数据。

numDocsWithField

包含当前域的文档的个数。

MinLength

记录所有域值长度最小的。

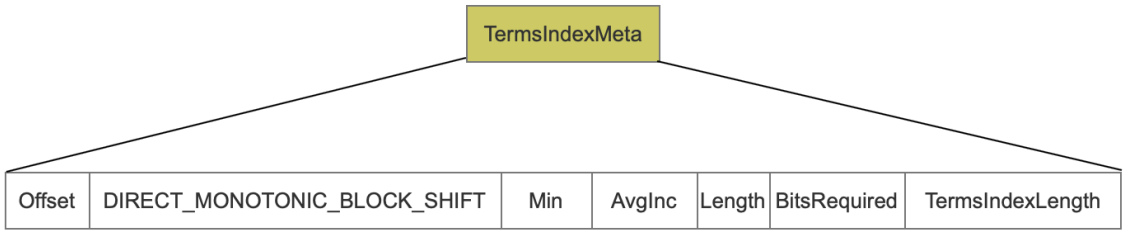
MaxLength

记录所有域值长度最大的。

记录MinLength跟MaxLength目的在于，如果两个值相同，那么就不需要.dvd文件中的TermsIndex字段、.dvm文件中的TermIndexMeta字段。因为在读取阶段，所有的域值的长度都相同，那么TermsValue中的数据是等分的。就很容易并且随机访问找到每篇文档中的域值。

TermsIndexMeta

图7：



Offset

TermsIndex数据段在.dvd文件的开始位置。

DIRECT_MONOTONIC_BLOCK_SHIFT

DIRECT_MONOTONIC_BLOCK_SHIFT用来在初始化byte buffer[]的大小，buffer数组用来存放每一个域值TermsIndex。

Min

记录一个最小值，在读取阶段用于解码。Min的含义请看我的源码注释。

AvgInc

记录一个AvgInc，在读取阶段用于解码。AvgInc的含义请看我的源码注释。

Length

在BinaryDocValues使用DirectMonotonicWriter的场景中，该值永远为0，不解释。

BitsRequired

经过DirectMonotonicWriter的数据平缓操作后，每个数据需要的固定bit位个数。

TermsIndexLength

TermsIndex数据段在.dvd文件的数据长度。

TermsIndexLength结合Offset，就可以确定 TermsIndex在.dvd文件中的数据区间。

结语

BinaryDocValues在处理域值时，不做跟ord值的映射，不做前缀存储，在特定场景下存储固定域值长度，域值长度较小时，查询域值速度优于SortedDocValues，排序性能接近SortedDocValues。 这是DocValues类型的关于.dvd、.dvm文件构造的最后一篇文章，我们只有在了解每一个DocValues的构造原理后，才能结合实际应用，使用合适的DocValues类型。

点击下载Markdown文件