

# SortedNumericDocValues

SortedNumericDocValues的索引结构跟NumericDocValues几乎是一致的，所以本文不会赘述跟NumericDocValues相同部分的内容，只介绍不同的部分数据结构。两种DocValue的最常用的使用场景就是对搜索结果进行排序，使用SortedNumericDocValues相比较NumericDocValues的优点在于，一篇文档中可以设置多个相同域名不同域值的SortedNumericDocValuesField，而NumericDocValuesField在一篇文档中只允许有一个相同域名的域。因此我们可以在不更改现有索引的情况下，只修改搜索的条件（更改Sort对象）就可以获得不同的排序结果，在以后介绍facet时会详细介绍这部分内容。

## 数据结构

### dvd

先给出NumericDocValues的.dvd文件的数据结构。图1：

#### .dvd (NumericDocValues)



再给出SortedNumericDocValues的.dvd文件的数据结构。图2：

#### .dvd (SortedNumericDocValues)



两个DocValues的DocIdData跟FieldValues部分的数据结构是一样的，因为源码中他们实际调用的是同一个方法来写入这两块的数据。下面介绍不同之处DocValueCount。

## DocValueCount

在上文中提到，索引阶段使用SortedNumericDocValues的话，一篇文档中可以有多多个相同域名不同域值的SortedNumericDocValuesField，而NumericDocValues只能有一个相同域名的NumericDocValuesField，如下图所示。图3：

```
// 1
doc = new Document();
doc.add(new NumericDocValuesField( name: "age", value: 92));
indexWriter.addDocument(doc);
```

图4：

```
// 1
doc = new Document();
doc.add(new SortedNumericDocValuesField( name: "age", value: 92));
doc.add(new SortedNumericDocValuesField( name: "age", value: 93));
indexWriter.addDocument(doc);
```

DocValueCount描述的信息即每篇文档中包含的相同域名不同域值的域的个数。

这些信息使用了DirectMonotonicWriter类进行了 趋势分解操作，然后使用PackedInts进行了压缩存储。DirectMonotonicWriter中的趋势分解的目的是尽可能减少空间的使用，它用来将 单调递增的整数序列(monotonically-increasing sequences of integers)进行平缓操作，使得在使用PackedInts进行压缩存储时，每一个数值能使用最少的固定bit位存储。

这里不赘述DirectMonotonicWriter中的趋势分解过程，可以看我的源码注释来理解这个过程：<https://github.com/luxugang/Lucene-7.5.0/blob/master/solr-7.5.0/lucene/core/src/java/org/apache/lucene/util/packed/DirectMonotonicWriter.java>。

## dvm

先给出NumericDocValues的.dvm文件的数据结构。图5：

.dvm (NumericDocValues)

Header	FieldNumber	DocvaluesType	DocIdIndex	NumValues	NumBitsPerValueMeteData	NumBitsPerValue	min	gcd	FieldValuesIndex	Footer
--------	-------------	---------------	------------	-----------	-------------------------	-----------------	-----	-----	------------------	--------

再给出SortedNumericDocValues的.dvm文件的数据结构。图6：

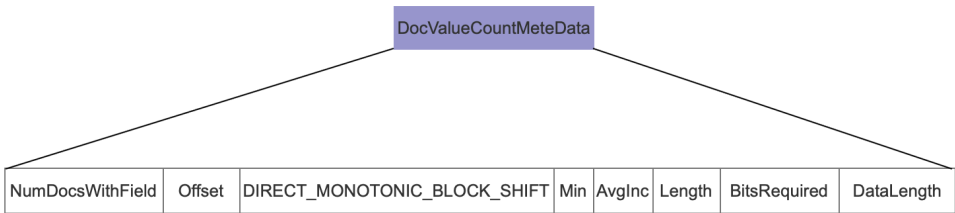
.dvm (SortedNumericDocValues)

Header	FieldNumber	DocvaluesType	DocIdIndex	NumValues	NumBitsPerValueMeteData	NumBitsPerValue	min	gcd	FieldValuesIndex	DocValueCountMeteData	Footer
--------	-------------	---------------	------------	-----------	-------------------------	-----------------	-----	-----	------------------	-----------------------	--------

上图中，除了红框标出的DocValueCountMeteData，其他信息都是与NumericDocValues一致的。

## DocValueCountMeteData

图7:



NumDocsWithField

NumDocsWithField描述了包含当前域的文档个数。注意的是如果.dvm中记录的NumValues值，描述的是当前域的域值个数，如果NumDocsWithField与NumValues的值相等，说明每篇文档中只有一个相同域名的SortedNumericDocValuesField，这种情况下就不用记录.dvd文件中的DocValueCount信息，并且此时NumDocsWithField跟SortedNumericDocValues在应用上几乎是一样的。

NumDocsWithField与NumValues相同的情况下，最终的.dvm文件如下图所示： 图8:

.dvm (SortedNumericDocValues)

Header	FieldNumber	DocvaluesType	DocIdIndex	NumValues	NumBitsPerValueMeteData	NumBitsPerValue	min	gcd	FieldValuesIndex	NumDocsWithField	Footer
--------	-------------	---------------	------------	-----------	-------------------------	-----------------	-----	-----	------------------	------------------	--------

Offset

Offset描述了DocValueCount信息在.dvd文件中的开始位置。

DIRECT\_MONOTONIC\_BLOCK\_SHIFT

DIRECT\_MONOTONIC\_BLOCK\_SHIFT用来在初始化byte buffer[]的大小，buffer数组用来存放每一篇文档中班包含的域值个数。

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SortedNumericDocValues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。

## DataLength

DocValueCount信息在.dvd文件中的数据长度。结合上面的Offset，在读取阶段，就可以确定应该读取.dvd文件中的某个数据区间，即DocValueCount信息。

## 结语

---

由于SortedNumericDocValues与NumericDocValues的索引文件数据结构非常类似，所以本篇介绍篇幅很小。SortedNumericDocValues这个名词中的Sorted的含义只有在一篇文档中包含多个相同域名不同域值的情况下才有价值体现。在以后介绍SortedNumericDocValues的应用时，会详细介绍它跟NumericDocValues的区别，本篇文章只是介绍在.dvd、.dvm文件中的索引数据结构。

[点击下载](#)Markdown文件