索引文件的生成（九）（Lucene 8.4.0）

上一篇文章中，我们介绍了在索引（index）阶段，Lucene收集了跟点数据相关的信息，这些信息在flush阶段会被读取，用于生成索引文件.dim&&.dii，从本文开始介绍索引文件.dim&&.dii生成的详细过程，如图1所示，另外阅读本文中需要前置知识：索引文件之dim&&dii：

图1：

MutablePointValues

MutablePointValues为图1流程图的准备数据，该对象中包含了在索引阶段收集的点数据的信息，我们在文章索引文件的生成（八）之dim&&dii中已经详细的展开介绍了，故这里只简单的列出这些信息：

- numPoints
- docIDs
- numDocs
- bytes

MutablePointValues对象中至少包含了上述的几个信息，但如果IndexWriter对象使用了IndexSort配置，那么MutablePointValues中还要额外包含一个信息：DocMap对象。

我们简单的回顾下IndexSort这个概念，文章构造IndexWriter对象（一）中我们说到，在构造一个IndexWriter对象的的过程中，其中一个流程是`设置IndexWriter的配置信息IndexWriterConfig`，当设置了IndexSort（IndexSort的一些介绍见文章文档提交之flush（三））配置后，段中的文档会按照IndexSort的排序规则进行段内的文档排序，由于每添加（例如IndexWriter.addDocument(...)方法）一篇文档，就排一次序，在实现（implementation）上不可能执行真正的数据排序（数据之间的交换），故通过一个映射关系，即DocMap对象来描述文档之间的排序关系，所以在一个段内，当设置了IndexSort配置后，每一篇文档有一个原始的段内文档号，该文档号按照文档被添加的先后顺序，是一个从0开始的递增值，而DocMap对象中提供了方法来描述文档之间基于IndexSort的排序关系，该方法在源码中的注释见 https://github.com/LuXugang/Lucene-7.5.0/blob/master/solr-8.4.0/lucene/core/src/java/org/apache/lucene/index/Sorter.java 中内部类DocMap提供的两个方法，这里简单的给出：

```
/* Given a doc ID from the original index, return its ordinal in the sorted index. */
    abstract int oldToNew(int docID);

/* Given the ordinal of a doc ID, return its doc ID in the original index. */
    abstract int newToOld(int docID);
```

执行处理前的初始化的工作

在当前流程点，我们就可以基于MutablePointValues中的信息来`执行处理前的初始化的工作`，工作内容为初始化以下几个信息：

- numLeaves
- splitPackedValues
- leafBlockFPs
- docsSeen
- parentSplits
- maxPackedValue
- minPackedValue

numLeaves

该值为long类型的变量，它用来描述我们随后即将构建的BKD树中的叶子节点的数量。

为什么能预先计算出BKD树中的叶子节点数量：

在图1的流程点`构建BKD树的节点值（node value）`中我们将会介绍Lucene将一个节点生成左右子树的划分规则（可以先看下文章Bkd-Tree简单的了解划分规则），该规则会使得总是生成一颗**满二叉树**，那么根据满二叉树的性质，我们只需要知道点数据的数量就可以计算出BKD树中的叶子节点的数量，而点数据的数量在收集阶段实现了统计，并且用numDocs（见上文中的MutablePointValues）来描述。

为什么要先计算出BKD树中的叶子节点数量：

在后面的流程中，我们可以根据当前处理的节点编号来判断当前节点是内部节点（inner node）还是叶子节点（leaf node），在介绍`构建BKD树的节点值（node value）`时会详细展开介绍numLeaves的作用，这里先介绍下节点编号是什么：

图2：

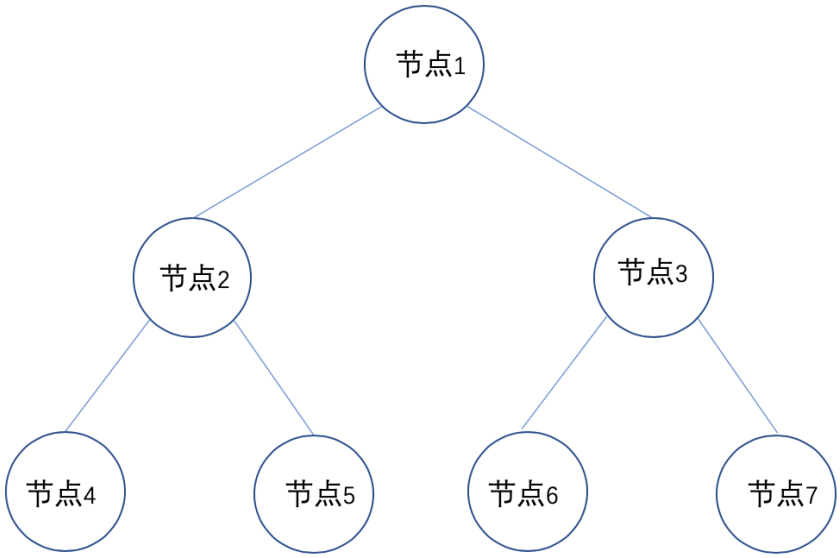


图2中，按照广度遍历的顺序，依次为每一个节点赋予一个节点编号，节点编号的作用在后面的流程中会使用到。

另外numLeaves也用来初始化例如splitPackedValues、leafBlockFPs，见下文。

splitPackedValues

splitPackedValues是一个字节数组，该值用来描述每一个节点使用哪个维度（维度编号）进行划分以及维度的值（在本篇文章中暂时不用理解这段话，在后面的文章中展开介绍），在当前流程点，我们只需要知道该数组的初始化的时机点，初始化的代码很简单，故直接给出：

```
final byte[] splitPackedValues = new byte[numLeaves * (bytesPerDim + 1)];
```

上述代码中，numLeavs为即将构建的BKD树中的叶子节点的数量，bytesPerDim的值为每个维度的值占用的字节数量，例如int类型的维度值占用4个字节（见文章索引文件的生成（八）之dim&&dii中关于数值类型转为为字节类型的介绍）。

leafBlockFPs

leafBlockFPs是一个long类型的数组，在当前流程点被初始化，如下所示：

```
final long[] leafBlockFPs = new long[numLeaves];
```

leafBlockFPs在随后的流程中会记录每一个叶子节点的信息在索引文件.dim中的起始位置，如下所示：

图3：

leafBlockFPs[ ]数组

图2中的LeafNodeData描述的是每个叶子节点的信息，可见leafBlockFPs数组中的数组元素数量为叶子节点的数量，即上文中的numLeaves。

docsSeen

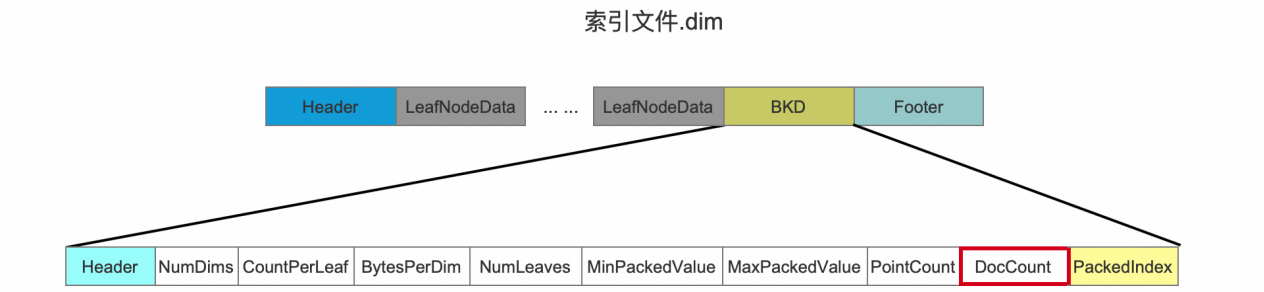
docsSeen是一个FixedBitSet对象，用来**去重记录**包含当前点数据域的文档的数量，例如我们添加下面两篇文档：

图4：

```
Document doc;
// 文档0
doc = new Document();
doc.add(new IntPoint( name: "content",  ...point: 3, 5, 12));
indexWriter.addDocument(doc);
// 文档1
doc = new Document();
doc.add(new IntPoint( name: "content",  ...point: 1, 5,23));
doc.add(new IntPoint( name: "content",  ...point: 3, 6,12));
indexWriter.addDocument(doc);
```

图3中，尽管有3条点数据内容，但是文档1中包含了2条，那么包含域名为"content"的点数据的文档的数量为2，docsSeen中统计的文档数量在后面的流程会被写入到索引文件.dim中，如下红框所示：

图5：



parentSplits

parentSplits是一个int类型的数组，首先看下初始化这个数组的源码：

```
final int[] parentSplits = new int[numDims];
```

上述源码中，numDims指的是当前处理的点数据的维度数，例如图3中处理的是三维的点数据，那么numDims的值为3。

在文章Bkd-Tree中介绍关于`选出切分维度`的内容时候说到，选择的判断依据如下：

```
条件1. 先计算出切分次数最多的那个维度，切分次数记为maxNumSplits，如果有一个维度的切分次数小于 (maxNumSplits / 2) ，并且该维度中的最大跟最小值不相同，那么令该维度为切分维度。
条件2. 计算出每一个维度中最大值跟最小值的差值，差值最大的作为切分维度(篇幅原因，下面的例子中仅使用了这种判定方式)。
```

我们只看条件一，parentSplits数组就是用来存储某个维度被选为切分维度的次数，在条件一中通过读取parentSplits数组来获得对应信息。

对于上述的两个条件，在后面的文章会再次提交，到时候再作详细的介绍。

maxPackedValue minPackedValue

maxPackedValue minPackedValue都是字节数组，在当前流程点`执行处理前的初始化的工作`中，通过遍历所有的点数据，找出每一个维度的最大值跟最小值，其中minPackedValue记录了每一个维度的最小值，maxPackedValue记录了每一个维度的最大值。

还是以图3为例，在遍历了3个点数据的信息后，maxPackedValue minPackedValue的数据如下所示：

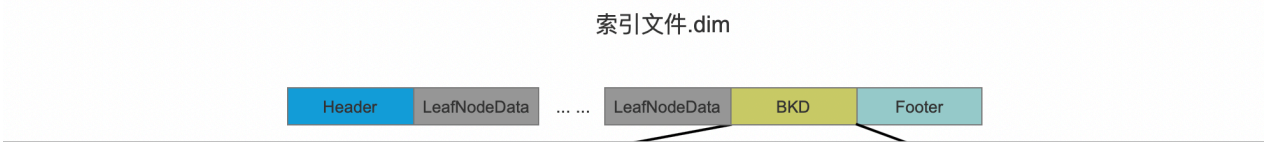
```
minPackedValue:{1, 5, 12}
maxPackedValue:{3, 6, 23}
```

为什么要统计maxPackedValue minPackedValue

上文中说到了`选出切问维度`的两个条件，其中条件2中，需要知道每一个维度中最大值跟最小值，而当前的maxPackedValue minPackedValue就用来为第一个节点的划分提供了依据。

在后面的流程中，maxPackedValue minPackedValue的值将会被记录到索引文件.dim中，如下红框所示：

图6：



结语

无。

点击下载附件