CHAPTER 4

4. QUERYING NON-SPATIAL DATA

4.1.INTRODUCTION

The execution performance of spatial queries such as range queries and KNN queries can be enhanced by using spatial indexes and colocating spatial files blocks that are accessed together. However, spatial indexes and spatial file colocation are not useful for non-spatial queries. If the query accesses some blocks using a non-spatial attribute such as “name”, all blocks are scanned to return the records that has “name” matching in that query.

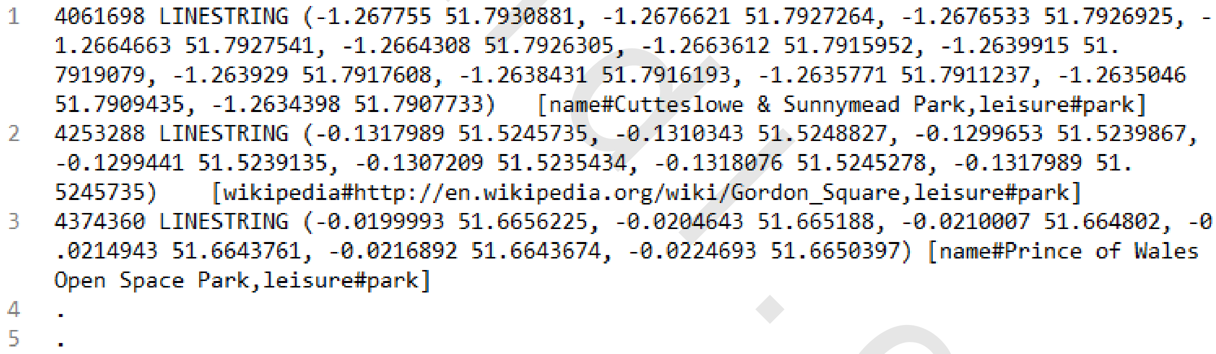
In [24], it is proposed to use inverted index [29] for indexing non-spatial attributes. We extend SpatialHadoop [1] by building a global inverted index. Global indexes are ones that are stored and maintained in the master node. Besides, we implement non-spatial operation to test inverted indexes.

4.2.INVERTED INDEX BUILDING

First, we need to choose the non-spatial attribute which is used to build this inverted file. There has been a huge research about choosing attributes based on the workload of queries, we can use any technique to choose the more effective attribute. We leave such procedures for future work

Co-SpatialHadoop implements a MapReduce job to build an inverted index on the chosen attribute using the following steps: (1) The Map function reads the partitioned blocks and checks records that use the chosen attribute. (2) It breaks the sentence assigned to the attribute of each record, and outputs every word as a key to a reduce function, and its block name as a value. (3) The output records that have a common key – or common word – are gathered to the same reduce function, which writes this key to the inverted index with all the blocks that contain this word. It also writes the occurrence of this word in every block. We can illustrate these steps using the following example.

OSM “parks” file prepared by SpatialHadoop has a collection of records, each record that represents park on the map has the following schema:



```
1   4061698 LINESTRING (-1.267755 51.7930881, -1.2676621 51.7927264, -1.2676533 51.7926925, -
    1.2664663 51.7927541, -1.2664308 51.7926305, -1.2663612 51.7915952, -1.2639915 51.
    7919079, -1.263929 51.7917608, -1.2638431 51.7916193, -1.2635771 51.7911237, -1.2635046
    51.7909435, -1.2634398 51.7907733)   [name#Cutteslowe & Sunnymead Park,leisure#park]
2   4253288 LINESTRING (-0.1317989 51.5245735, -0.1310343 51.5248827, -0.1299653 51.5239867,
    -0.1299441 51.5239135, -0.1307209 51.5235434, -0.1318076 51.5245278, -0.1317989 51.
    5245735)    [wikipedia#http://en.wikipedia.org/wiki/Gordon_Square,leisure#park]
3   4374360 LINESTRING (-0.0199993 51.6656225, -0.0204643 51.665188, -0.0210007 51.664802, -0
    .0214943 51.6643761, -0.0216892 51.6643674, -0.0224693 51.6650397) [name#Prince of Wales
    Open Space Park,leisure#park]
4   .
5   .
```

Figure 3.10: Some records of Parks file

Each record consists of: (1) park ID, (2) vector of points to draw the outline of the park on the map, and (3) number of text attributes, such as: name, leisure and wikipedia.

If the chosen attribute of the inverted index operation is “name”, the map function chooses records 1 and 3 because they have the “name” attribute. Then, it breaks attribute values to words, each word is the key of a reduce function and the value is block name which contains that record. The reduce phase collects each key in different reduce function, and it outputs the key, the contained blocks, and the number of occurrences of this key in this block to the inverted file. If the first record is located in block1 and the third record is located in block2, the output of the reduce phase for the above example is:

Cutteslowe => block1:1
Sunnymead=> block1:1
Park=> block1:1, block2:1
Prince=> block2:1
of => block2:1
Wales => block2:1
Open => block2:1
Space => block2:1

In our work, we have extended SpatialHadoop by non-spatial operation. Using the above inverted index, if we query the spatial file for records that have the attribute "name" with the value "Space", block2 only is scanned instead of the two blocks, and the time needed by scanning is reduced to 50%.

4.3. CONCLUSION

We propose using inverted index of non-spatial attributes in the data to enhance the performance of executing non-spatial queries. We implement MapReduce job to index spatial files using these non-spatial attributes and generate inverted files indexes to be used by non-spatial queries.