CHAPTER 1

1. INTRODUCTION

The rapid growth of using spatial data by many systems such as GIS and LBS is driven by the development of new smart platforms that provide efficient solutions for the challenges of spatial data.

Spatial queries executed by these systems are complex and require high server capabilities because they frequently access large data sets and use them in intensive computations. Moreover, these types of queries generated by these applications need to processing on highly scalable platforms that are capable of executing the queries in near real-time.

Hadoop is highly scalable platform that provides fault tolerance mechanism and is used in distributed processing of large-scale data sets. However, it needs to be extended to support some spatial features to efficiently process spatial data. This thesis introduces Co-SpatialHadoop – spatial extension for Hadoop platform – that colocates spatial data to obtain enhancement in network usage. It also builds non-spatial indexes to enhance response time of non-spatial operations.

This chapter is organized as follows. First, we discuss the motivation for this work and the scope of this thesis in Section 1.1. Next, the main contributions of this thesis are listed in Section 1.2. Finally, the organization of the thesis is presented in Section 1.3.

1.1. MOTIVATION AND SCOPE OF WORK

The recent evolution of ubiquitous positioning technologies has motivated the rapid growth of spatial data. Special sensors, satellites and GPS devices capture spatial data to be used in several domains, such as scientific researches and business areas. Figure 1.1 shows the main categories of these domains, which are: LBS and "Pathology services". LBS have many branches such as business, social networks and maps.

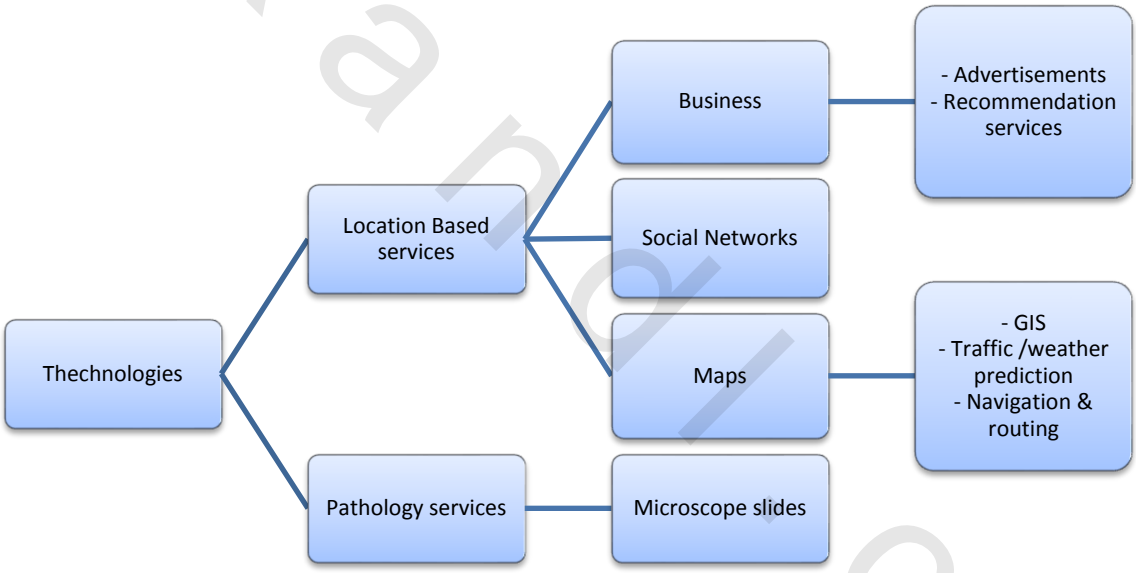

Figure 1.1: Technologies and services that use spatial data

LBS are used in many business applications such as advertisements and recommendations services. For example, an application can keep track of the current locations of its users; a restaurant can send its offers or advertisements to customers who are within a radius of 100 kilometers from the restaurant. LBS can additionally help social scientists to study the dynamics of social networks and understand human behavior. Normal peo-

ple use location based services provided by various social network applications to share their locations.

Moreover, LBS are used by geographic information systems (GIS). GIS providers collect and manipulate huge spatial data sets to use them in (1) web mapping service applications, (2) generating spatial reports and statistics for instance about traffic incidents or (3) weather applications such as those used for hurricane path prediction. This thesis uses real GIS spatial data in the experiments to evaluate Co-SpatialHadoop performance.

The second important category of applications that heavily rely on spatial data processing is scientific technology. “Pathology image analysis” is an example. It scans microscope slides and translates them to spatial data in order to be used later in heavy computations to detect diseases.

The various applications discussed above execute various types of spatial queries that access huge streams of spatial data to perform complex and intensive computations. In business, KNN (k nearest neighbors) queries and range queries are commonly used for geo-marketing and recommendation services. KNN queries determine the k nearest neighbors of a given position and range queries determine all the users located in a given range. Social scientists perform complex analysis operations on spatial data collected about residents of a certain area to produce reports, graphics, and statistics. In GIS systems, it is common to use spatial join queries to join multiple sets of spatial objects to show them as layers on a map. A common challenge in all these applications is executing these spatial queries in near real-time.

MapReduce and its open source implementation Hadoop have emerged as a scalable and cost effective distributed computation model. However, Hadoop is unaware of spatial data properties and operations. Spatial queries access data for a specific area, yet HDFS by default scans the entire data. Therefore, Hadoop is not suitable for executing spatial queries which can benefit from using an index to select the data in a specific area.

Additionally, spatial data has multi-dimensional nature; these dimensions are geographic coordinates such as X and Y axis, in addition to time dimension. LBS user for example has dynamic geographic location with time, so its records have three dimensions: X,

Y and time. This multi-dimensional nature cannot fit with key-based partitioning nature of HDFS that categorizes data sets using one key for every data set.

Several spatial approaches in literature have been proposed for adding support to spatial data and queries to Hadoop. These approaches can be categorized as follows: (1) adding new layers to the Hadoop stack to support spatial indexes for data stored in HDFS and processed by Hadoop. The main goal of this approach is to improve accessing spatial data, (2) extending the dataflow layer of the Hadoop stack, which allows compiling queries written in high level languages to Hadoop job to support spatial data type and spatial primitive functions and operation.

HDFS distributes data partitions and replicas among nodes using *a default placement policy* to guarantee data availability and load balance. However, this placement policy is unaware of spatial data properties. To achieve a better performance at executing spatial queries, it is important to colocate related data partitions - which can be processed together - in the same nodes of HDFS. Deciding which partitions to colocate is passed to the spatial properties of the data. This is expected to improve the execution time of spatial queries due to decreasing the network overhead and data transmissions between nodes. Related works have not studied colocating spatial data. This thesis introduces Co-SpatialHadoop, which examines colocating data based on their spatial properties and proposes *new placement policy* for HDFS to accomplish this.

1.2. MAIN CONTRIBUTIONS

- Extend HDFS with a new block placement policy which colocates data partitions based on their spatial properties. Our main goal is to minimize the need for transferring blocks accessed by a map task through the network. This is done through colocating all the blocks that a map task access on the same node that it will be executed.
- Provide HDFS and SpatialHadoop [1] index layer (R-Tree) the suitable updates to integrate with the new block placement policy.
- Create inverted indexes on non-spatial attributes of the data sets, to enhance the performance of non-spatial queries.
- Add non-spatial “select” operation to the operations layer of SpatialHadoop.

- Do extensive experiments that use datasets of world cities, streets, rivers, and lakes obtained from OpenStreetMap (OSM).

1.3. THESIS ORGANIZATION

The remainder of this thesis is organized as follows. In Chapter 2, we present the required background, terminology used in this thesis and a summary of research contributions to this work. In Chapter 3, we describe our proposed system architecture for Co-SpatialHadoop. First, we will describe the new components that needs to be added to the Hadoop stack to add support for spatial data and queries, and to colocate data partitions based on their spatial properties. Next, we discuss adding indexes on non-spatial attributes to support a wider range of queries. In Chapter 4, we present an extensive set of experiments that compares the performance of Co-SpatialHadoop with SpatialHadoop to show the benefit of adding our proposed features. Finally, we summarize the thesis contributions and present possible extensions to the proposed work in Chapter 5.