BSDM: Big Spatial Data Management

Special Track along with CLOUD COMPUTING 2017, February 19 - 23, 2017 - Athens, Greece
http://www.iaria.org/conferences2017/CLOUDCOMPUTING17.html

Michael Vassilakopoulos
Data Structuring & Engineering Lab
Dept. of Electrical & Computer Eng.
University of Thessaly
Volos, Greece
mvasilako@uth.gr

***Abstract*—We are living in the era of Big Data. Spatial and Spatiotemporal Data are not an exception. Mobile apps, cars, GPS devices, UAVs, ships, airplanes, space telescopes, medical devices and IoT devices are generating explosive amounts of data with spatial characteristics. Web apps and social networking systems also store vast amounts of geo-located information, like geo-located tweets, or captured mobile users' locations. Modeling, storing, querying and analyzing big spatial and spatiotemporal data is an active area of basic and applied research with many challenges. Multicore CPU / GPU processing techniques and parallel and distributed frameworks utilizing cloud infrastructures are being created and extended for novel big spatial data management solutions. The purpose of this track is to act as a forum where recent advances in Big Spatial (and Spatiotemporal, considered as a special case of Spatial) Data Management will be presented and discussed.**

Keywords-Big data; Spatial data; Cloud computing; Data modelling and analysis; Data processing; Systems and applications; Algorithms.

I. INTRODUCTION

In today's world, vast amounts of data are being created from numerous applications and sources (e.g., sensor data, archives, docs, business apps, web, social media). The term *Big Data* is used to denote such data. Big data refers to data sets that are too complex, or voluminous for traditional data management systems to handle.

Spatial Data is data expressing the geographic features of objects and elements on, below, or above the earth's surface. Such data appear in geography related applications, e.g., Geographic Information Systems (GIS), astronomy, environmental monitoring, earthquake research, weather forecasting, traffic management. However, spatial are also multidimensional data and this term is used extendedly to express multidimensional data from several application domains beyond geography, like medicine, or biology.

Mobile apps, cars, Global Positioning System (GPS) devices, Unmanned Aerial Vehicles (UAVs), ships, airplanes, telescopes, medical devices and IoT devices are generating explosive amounts of data with spatial characteristics. Web apps and social networking systems also store vast amounts of geo-located information, like geo-located tweets, or captured mobile users' locations. We are living in the era of big spatial data [1]. Capturing the evolution of spatial data in time gives rise to an even more complex concept, *Spatiotemporal Data*. Due to their multidimensional nature, spatial / spatiotemporal data are harder to handle than data in traditional applications (e.g., names, numbers, dates, etc.).

Data Management is a broad term used to describe practices, systems and theories for properly managing data of an enterprise, scientific, or social activity. Within this term, models, architectures, systems, analysis, design, storage, querying, mining, security, quality, governance and integration involved throughout the data lifecycle are included. *Spatial Data Management* is the extension of data management to the more complex spatial-data universe. *Big Spatial Data Management* is even more demanding and requires the exploitation of modern computing features.

In stand-alone systems, the exploitation of large amounts of main memory, Solid-state Drives (SSDs), multiple cores of Central Processing Units (CPUs), or the massively parallel architecture of Graphics Processing Units (GPUs) [2] can be used for big spatial data management. Parallel and distributed computing using shared-nothing clusters for big data management is a research trend during last years. *MapReduce* is a programming model suitable for such clusters and *Apache Hadoop* [3] is a popular open-source software framework using this model. *Apache Spark* [4] is another, more recent, open-source cluster-computing framework, developed to overcome limitations of Apache Hadoop. Such systems are usually implemented within a *Cloud Computing* environment and, beyond processing speed, they provide failure resilience and scalability. Several spatial extensions of Hadoop (e.g., Hadoop-GIS [5] and SpatialHadoop [6]) and Spark (e.g., SpatialSpark [7], LocationSpark [8], SIMBA [8]) have appeared during last years. Each of these systems supports storage and querying of big spatial data. However, these systems have significant differences regarding their supporting distributed computing frameworks, data models, programming languages, geometric and spatial processing Application Programming Interfaces (APIs) and algorithms utilized for data processing. Studying their relative performance, as well as, enhancing their cababilities is an active area of research.

II. Submissions

The first paper of the track is a joint effort of George Mavrommatis, Panagiotis Moutafis and myself. It is entitled "Closest-Pairs Query Processing in Apache Spark" and is related to efficient query processing in a parallel and distributed framework. More specifically, the (K) Closest-Pair(s) Query (KCPQ) is studied. This query consists in finding the (K) closest pair(s) of objects between two spatial datasets. This query is among the popular ones in spatial data processing and its computation is demanding, since all the possible pairs that can be formed between the two spatial datasets are candidates for inclusion in the final result. In this paper, processing of this query in Apache Spark is presented. The presented algorithm separates data in strips and utilizes the plane sweep technique (a technique well known in computational geometry) within each strip and between strips. An experimental analysis of the performance of this algorithm, based on big real-world datasets, is also included. The future plans of the authors include the elaboration of this algorithm to reduce the network communication traffic within the distributed framework, the calculation of an improved initial bound for faster processing of data, the comparison of the performance of this algorithm against other solutions working in parallel and distributed environments and the study of its scalability.

The second paper of the track, entitled "A Raster SOLAP Designed for the Emergency Services of Brussels Agglomeration", is authored by Jean-Paul Kasprzyk and Jean-Paul Donnay. It presents the design and implementation of a Spatial On-line Analytical Processing (SOLAP) system for decision making in emergency services of Brussels agglomeration. To quickly reach incident locations, emergency services have to fairly distribute their resources in the area, based on analysis of risk data, infrastructure information and socio-economic factors. The system developed allows decision-makers to generate risk maps and to compare them with the accessibility of resources. Moreover, it provides the capability to perform simulations on resource locations and test their impact on accessibility. This system is based on the raster model (a model best suited for the representation of data which are continuous in space). Following a state-of-the-art section, where risk analysis and SOLAP tools are reviewed, the architecture of the system developed is presented in detail. The implementation of this system with open-source tools and its interface are presented in the sequel. The authors mention as further improvements the inclusion of data about the speed of roads that depend on hour of the day and day of the week and the inclusion of the durations of historical interventions from historical data.

The third paper of the track is authored by Maria Koziri and Thanasis Loukopoulos. It is entitled "Sensor Selection for Resource-Efficient Query Execution in IoT Environments" and presents an algorithm for selecting a subset of sensors that are distributed in space for answering a spatial query with adequate accuracy of the result, while minimizing the total energy consumed. Such sensors may exist in an Internet of Things (IoT) environment and can scale to the orders of millions or even billions, especially in future and emerging applications. Sensors might operate on battery, therefore, reducing energy consumption can extend their lifetime. This paper presents a rigorous problem formulation that expresses the trade-off between increasing quality of query results and resource consumption. Solving this problem can be shown to be NP-hard. Therefore, heuristics could be used to compute optimized solutions. Next, a greedy algorithm is presented. This is a two-step algorithm. In the first step, it covers one of the constraints expressed in the problem formulation and in the second step it iteratively optimizes the solution. This algorithm is experimentally compared against random assignment of sensor subsets. The results indicate that the greedy alternative leads to significant improvements in relation to random assignment, regarding resource consumption.

III. Conclusion

The BSDM special track includes three papers that relate to different topics among the broad range of topics included in Big Spatial Data Management. The audience will be informed about arising algorithms, applications, systems and theory in this thriving research domain.

Acnowledgements

I would like to thank the organizers of CLOUD COMPUTING 2017 for their tireless efforts and for accepting BSDM as a special track. I also thank the reviewers for their informative feedback. Last, but not least, I am thankful to the authors for their interesting contributions.

References

[1] A. Eldawy and M.F. Mokbel, "The Era of Big Spatial Data: A Survey," Foundations and Trends in Databases, vol. 6, no 3-4, pp. 163-273, 2016.

[2] J. Zhang, S. You, and L. Gruenwald, "Large-scale spatial data processing on GPUs and GPU-accelerated clusters," SIGSPATIAL Special, vol. 6, no 3, pp.27-34, 2015.

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," OSDI 2004, pp. 137-150, 2004.

[4] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, pp. 10–10, 2010.

[5] A. Aji, et al., "Hadoop-GIS: A high performance spatial data warehousing system over MapReduce," PVLDB, vol. 6, no. 11, pp. 1009-1020, 2013.

[6] A. Eldawy and M. F. Mokbel, "SpatialHadoop: A MapReduce framework for spatial data," ICDE 2015, pp. 1352-1363, 2015.

[7] S. You, J. Zhang, and L. Gruenwald, "Large-scale spatial join query processing in Cloud," ICDE Workshops 2015, pp. 34-41, 2015.

[8] M. Tang, Y. Yu, Q. M. Malluhi, M. Ouzzani, and W. G. Aref, "LocationSpark: A distributed in-memory data management system for big spatial data," PVLDB vol. 9, no. 13, pp. 1565-1568, 2016.

[9] D. Xie, et al., "Simba: Efficient in-memory spatial analytics," SIGMOD Conference 2016, pp. 1071-1085, 2016.