SCALABLE ANALYSIS AND RETRIEVAL OF POLARIMETRIC SAR DATA ON ELASTIC COMPUTING CLOUDS

Luigi Mascolo, Pietro Guccione

Department of Electrical and Information Engineering, Polytechnic of Bari, Via E. Orabona 4, 70125 Bari, Italy

Marco Quartulli, Igor G. Olaizola

Vicomtech–IK4, Mikeletegi Pasealekua 57, Parque Tecnológico, 20009 Donostia–San Sebastian, Spain

Giovanni Nico

Istituto per le Applicazioni del Calcolo, National Research Council, Via Amendola 122, 70126 Bari, Italy

ABSTRACT

Earth Observation (EO) mining systems aim at supporting efficient access and exploration of large volumes of image products. In this work, we address the problem of content-based image retrieval via example-based queries from Petabyte-scale EO data archives. To this end, we propose an interactive data mining system that relies on distributing unsupervised ingestion processes onto virtual machine instances in elastic, on-demand computing infrastructures that also support archive-scale content indexing via a "big data" analytics cluster-computing framework. In particular, we focus on the analysis of polarimetric SAR data, for which target decomposition theorems have proved fundamental in discovering patterns in data and in characterizing the ground scattering properties. Experiments are carried out on the publicly available UAVSAR full polarimetric data archive, whose basic products amount to about 0.64 PB of storage. We report the results of the tests performed by using a public IaaS. The obtained measures appear promising for data mapping and information retrieval applications.

Index Terms— Content–Based Retrieval, Remote Sensing, Elastic Cloud Computing, Big Data, Polarimetric SAR

1. INTRODUCTION

Remotely sensed data volumes are growing at faster and faster rates due to the increasing number of spaceborne and airborne Earth Observation (EO) missions and to the tightening of the image resolution requirements. As an example, according to the last yearly report published at the end of 2013, the archives of NASA's distributed Earth Observing System Data and Information System (EOSDIS) amount to almost 10 petabytes, with 6,900 accessible datasets and an average archive growth of approximately 8.5 terabytes per day.

The capability of both accessing and processing such large data volumes represents the raw matter that allows planetary-scale applications like deforestation monitoring, glacial retreat investigation, urban development mapping, land cover classification and so on. The difficulties of mining EO imagery archives are reflected in the fact that most of the images have never been seen and least of all analyzed by a human analyst. Allowing an efficient discovery, annotation and retrieval of data products is the goal of EO mining systems [1]. However, current approaches for image mining fall short of the efficiency requirement due to bottlenecks in their infrastructure that prevent the fulfillment of their main goal.

In order to address the problem, approaches to manage "big data" are being proposed. Among the several possible computing approaches, MapReduce is probably the most popular paradigm and it has already proved successful in processing massive amounts of data, relying on commodity machines [2]. Recent works present frameworks based on such paradigm to analyze and retrieve information from large data archives [3]. For example, in [4] the authors propose a system to perform data analytics on the archives produced by the European Space Agency's Gaia mission. However, the MapReduce computing paradigm lacks of the flexibility required for a query–by–example retrieval system and several weaknesses have been pointed out in literature [5]. Among them, there are limitations in efficiently expressing iterative computations that are typical of the machine learning tasks employed for the generation of tree-structured archive content indices.

In this paper, we present an EO data analytics tool that allows the exploration of massive datasets via example-based queries. The tool is both efficient and fault-tolerant and is based on an open-source cloud-computing engine for big data analytics, Apache Spark [6], [7], and on the on-demand cloud-computing infrastructure offered by the Amazon Elastic Compute Cloud (EC2). The system can be exploited to interactively query large-scale EO imagery archives with response times in the order of the second. We focus on Synthetic Aperture Radar (SAR) data, where the improvements in the sensor resolution and the use of multiple polarizations are causing a relevant growth of data volume.

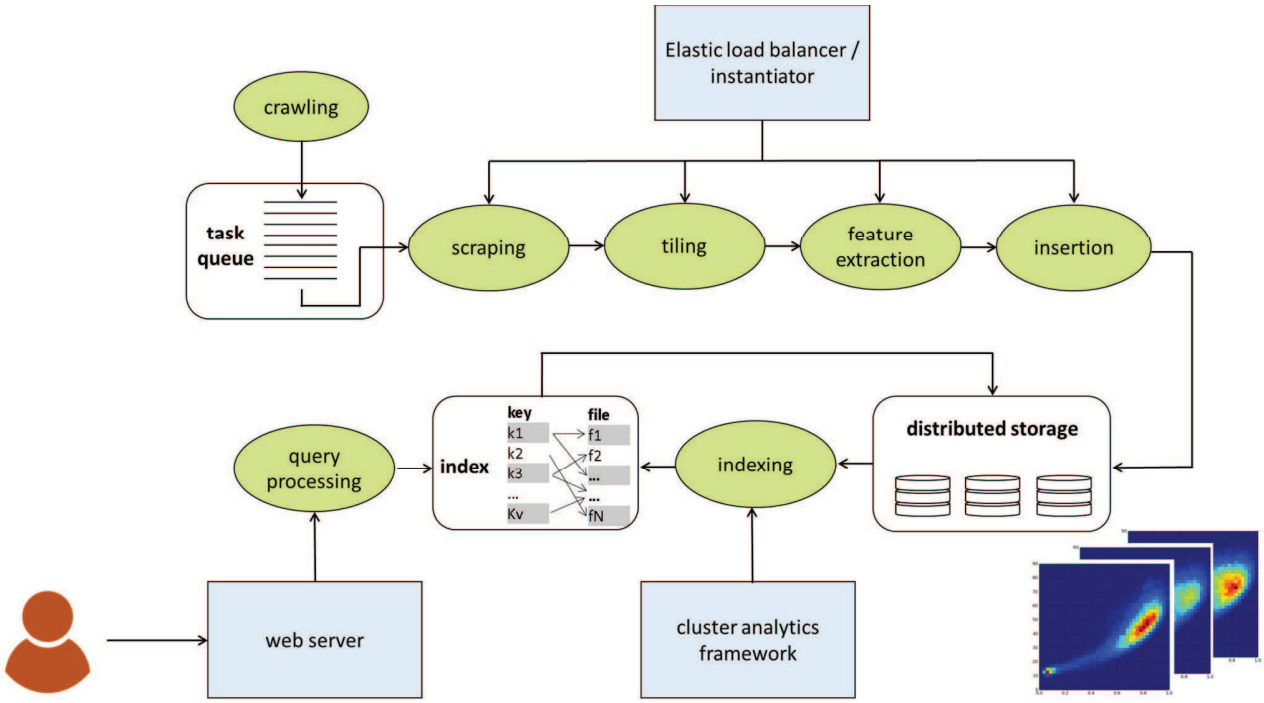

Fig. 1. A sketch of the high level architectural decomposition of the system. In the ingestion phase, crawlers populates a task queue with identifiers of the products to be ingested. One or more workers instantiated by an elastic load balancing mechanism access the data products (scraping), divide them into tiles, extract descriptors for each of them (feature extraction) and insert these descriptors into a distributed storage database. In the indexing phase, a cluster of machines is orchestrated by a cluster analytics framework to build a content-based index of the items in the distributed storage database. The results of the indexing process consist of a data structure that allows logarithmic-complexity content queries to take place. They are made available to a query processing system that is invoked via a web server by the operations performed by a user in an interactive visualization subsystem.

Experiments are carried out on publicly available UAVSAR data archives and the performance is quantitatively evaluated.

This contribution is structured as follows. In Section 2, a description of the main parts of the proposed content-based retrieval system for polarimetric data is given. In Section 3, we provide a brief description of the dataset and present preliminary performance measured during the analysis of massive repositories. The conclusions close the paper.

2. SYSTEM FRAMEWORK AND ARCHITECTURE

The potential of SAR polarimetry to characterize the physical properties of the Earth surface has led to a variety of applications that aim at exploiting the scattering mechanism of the ground to extract geophysical parameters and perform landcover classification. Ground targets can be characterized based on the way they scatter the electromagnetic waves, as described by the so-called scattering matrix. Currently, Cloude–Pottier's decomposition is probably the mostly used method for target decomposition and it relies on the eigen-analysis of the target coherency matrix [8].

In this work, we exploit Cloude–Pottier's theory to estimate three variables of interest, the mean α angle, the entropy H and the anisotropy A. Basing on these measurements, retrieval of polarimetric SAR images can be carried out in a completely unsupervised way. Indeed, such descriptors intrinsically define the classes to which the targets belong, on the basis of the physical properties (single, volume or double– bounce scattering), described by α, and of the degree of statistical disorder (pure or distributed target), described by H and A. In this Section we aim at providing a global description of a cloud-based retrieval system for polarimetric SAR images.

2.1 System Architecture

A sketch of the main parts of the proposed content-based retrieval system is illustrated in Fig. 1. It is composed by four main processing subsystems: the ingestion, the indexing, the search and the interactive visualization subsystem.

The ingestion subsystem is the part of the system concerned with ingesting and processing operations on data

products. A crawler agent is charged with performing web page analysis with the aim of discovering new data products and to input the results into a queue system. Then, a load balancer is designated to manage resource usage on an elastic cloud infrastructure, by adaptively instantiating virtual ingestion machines based on the current needs, in order to scrape data products from web pages and performing ad-hoc processing procedures [9]. The processing operations consist of two main tasks: tiling the images in geo-referenced patches, whose size depends on the kind sensor, on the resolution and on the aims of the content-based searches, and performing feature extraction operations. The images descriptors and metadata are finally pushed to a distributed, scalable and fault-tolerant storage system.

The indexing subsystem makes use of a cluster analytics framework to produce one or more indexes of the items stored in the database, in order to allow efficient subsequent searches with logarithmic complexity. Typically, machine-learning algorithms are involved in the index creation and, in general, such algorithms have been designated to run locally, on a single machine. However, performing analysis of large-scale data by a single machine is impractical due to capacities limitations. Cluster analytics frameworks are suitable to develop parallel iterative algorithms, like those required by the indexing subsystem, and that, in addition, have to be executed on multiple machines using elastic computing platforms.

The search subsystem is concerned with transforming the user provided examples to a form suitable for query processing, i.e. performing the processing and feature extraction operations on them, and then to determine the best matching tiles based on the content index.

Finally, an interactive visualization subsystem provides the user with a query interface with the search engine and a visualization module to display the results.

3. LARGE SCALE SYSTEM OPERATIONS

For the experiments, we used the full polarimetric SAR data products from the UAVSAR public image archives. Each polarimetric acquisition consists of six images, i.e. the cross-products of the four Single Look Complex files corresponding to the measurement of the scattering matrix Shh, Shv, Svh and Svv. In this work, the ground-projected polarimetric products have been considered (equiangular geographic projection, 6-by-6 m pixels resolution).

3.1 Ingestion

For the ingestion phase, an elastic cluster of machines has been instantiated on the cloud-computing infrastructure offered by the Amazon Web Services (AWS) Elastic Compute Cloud (EC2), with the task of analyzing data from the publicly available NASA's Jet Propulsion Laboratory

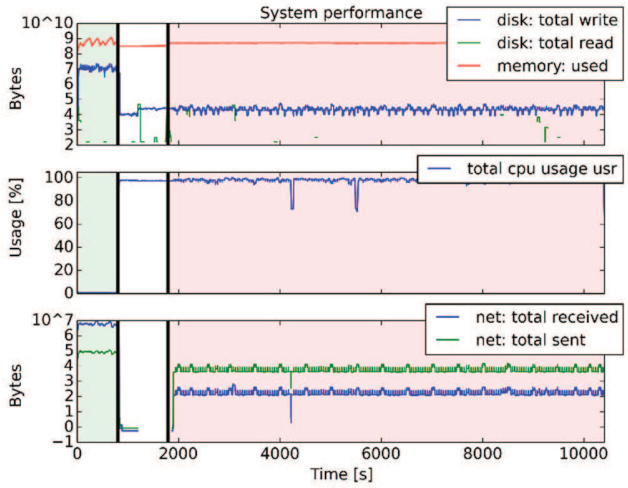

Fig. 2. Graphs of the system performance during the ingestion phases: scraping (shaded green), tiling (white) and feature extraction / insertion (shaded red). The performance of disk and memory usage (upper plot), of total cpu usage (middle plot) and of total network traffic (lower plot) are shown.

(JPL) archives. During ingestion, each image product is divided in 500-by-500 pixel sized square patches, corresponding to approximately 3-by-3 km areas. Such dimension is chosen in order to both preserve locality for retrieval and to retain enough information for the subsequent feature extraction. In Fig. 2, we report the plots of system performance parameters during the ingestion phase for one of the ingestion nodes instantiated on AWS-EC2. It has to be observed that the scraping phase requirements are costly in the sense of network transfers because large data products have to be transferred from their storage location to the cluster entities location. Computational and memory resources are significantly employed for the tiling and feature extraction tasks. In particular, for the latter task, computing capacities are requested for pixel-level eigen-decomposition. Finally, for the insertion tasks that run in parallel with feature extraction, network resources are needed for loading data in the distributed storage system (HDFS, in our case).

At the time of writing, approximately 0.65 TB of data have been collected. Currently, the data collection and processing operations are an ongoing effort and it is expected for the system to process and ingest tile descriptors of about 0.1% of the total database at conference time.

3.2 Indexing

A parallelized version of the tree structured vector quantization algorithm has been implemented for the index construction. The algorithm is iterative and grounds on a parallel implementation of the k-means algorithm that is

suitable to operate on very large volumes of data. The objective is to learn a partition-by-similarity of the data space, corresponding to a binary tree structure, thereby allowing for logarithmic complexity query-processing operations. The algorithm is described in terms of iterations consisting of multiple calls to user defined Map and Reduce functions, according to the MapReduce programming model [2].

MapReduce is suitable to develop parallel applications that have to be executed on large clusters of commodity machines. In general terms, the Map function takes an input key/value pair and produces a set of intermediate key/value pairs. The Reduce function accepts an intermediate key and the values associated with it and merges the values to form a possibly smaller set of values.

In addition to the fault-tolerance expected by a standard big data processing system capable of running on ad-hoc clusters of machines, since the archive content indexing system is based on iterative computations, it is important for the underlying framework to support in-memory computations performing repeated queries on a subset of data. It is evident that efficiently reusing intermediate results across multiple computations becomes a fundamental feature in this context. We build the indexing subsystem on top of Apache Spark, a general-purpose framework for large-scale distributed data processing, based on the abstraction of Resilient Distributed Datasets (RDDs) [6], [7]. RDDs are immutable collections of objects scattered on a set of machines that are built through parallel transformations. Two kinds of operations can be performed on RDDs: transformations and actions. Transformations generate a new dataset from the input one, for example by performing mapping or filtering operations, whereas actions, like reduce, count or collect operations, return a value or export data to a storage system.

The query processing subsystem operates on this index and is able to retrieve efficiently a small subset of the most similar tiles with respect to a user-provided example. We observed response times of the order of the second during repeated database queries.

CONCLUSIONS

In this work, we developed a complete system for content-based retrieval of SAR polarimetric data. The system has two main elements of novelty: it makes use of public cloud based services and of a cluster-computing platform to perform large-scale data analysis. The system has proved efficient to mine large volumes of data archives. Experiments have been carried out on a small percentage of the entire JPL's UAVSAR imagery archives, which consist of approximately 0.64 PB of data.

Although developed for polarimetric SAR data, the system schema is general and can be further extended to include data from other sensor, thereby allowing efficient searching across heterogeneous geospatial repositories. In particular, an application to mine full Sentinel-1 archives can be devised by considering a quasi-complete description of the radar signal in terms of textural and shape descriptors as well as more advanced descriptions for metric resolution data based on signal decomposition in fractional frequency domains [10][11].

ACKNOWLEDGEMENTS

The authors would like to thank NASA's Jet Propulsion Laboratory for providing availability and free access to data products. The research was partially supported by a CNR (Italian National Research Council) scholarship (Short Term Mobility Program, 2014).

REFERENCES

[1] M. Quartulli and I. Garcia Olaizola, "A review of EO image information mining," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 75, pp. 11–28, 2013.

[2] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107– 113, 2008.

[3] D. Moise *et al.*, "Terabytescale image similarity search: experience and best practice," in Big Data, 2013 IEEE International Conference on. IEEE, 2013, pp. 674– 682.

[4] D. Tapiador *et al.*, "A framework for building hypercubes using MapReduce," Computer Physics Communications, vol. 185, no. 5, pp. 1429–1438, 2014.

[5] C. Doulkeridis and K. Nørvåg, "A survey of large-scale analytical query processing in MapReduce," The VLDB Journal, vol. 23, no. 3, pp. 355– 380, 2014.

[6] M. Zaharia *et al.*, "Spark: cluster computing with working sets," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010, pp. 10–10.

[7] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 2–2.

[8] S. R. Cloude and E. Pottier, "A review of target decomposition theorems in radar polarimetry," Geoscience and Remote Sensing, IEEE Transactions on, vol. 34, no. 2, pp. 498–518, 1996.

[9] N. R. Herbst, S. Kounev and R. Reussner. "Elasticity in Cloud Computing: What It Is, and What It Is Not." ICAC. 2013, pp. 23-27.

[10] M. Walessa and M. Datcu. "Model-based despeckling and information extraction from SAR images." Geoscience and Remote Sensing, IEEE Transactions on, vol. 38, no. 5, pp. 2258-2269, 2000.

[11] J. Singh, M. Datcu, "SAR Image Categorization With Log Cumulants of the Fractional Fourier Transform Coefficients," Geoscience and Remote Sensing, IEEE Transactions on, vol.51, no.12, pp.5273-5282, 2013.