DISTRIBUTED THEMATIC MAPPING PERFORMANCE OPTIMIZATION IN PUBLIC CLOUDS

J. Lozano, M. Quartulli, J. Egaña, I.G. Olaizola

Vicomtech-IK4
Digital Media Department
Paseo Mikeletegi 57,
20009 Donostia, Spain

E. Zulueta

University of Basque Country
Sys. Eng. and Automation Department
Nieves Cano 12,
1006 Gasteiz, Spain

ABSTRACT

Global distributed thematic mapping in public clouds requires optimized data flows. These optimized flows can be the result of the analysis by Machine Learning (ML) of a deeply sensorized mapping system. In this sense, distributed global mapping requires a monitoring system that allows to understand the internal working of the system and enables the implementation of corrective actions to increase system performance. This work presents an implementation of a system monitoring framework and the obtained analysis results.

***Index Terms*—** System monitoring, big data, web mapping

1. INTRODUCTION

Performance-optimized tasks are required to make the best use of large scale geospatial computing infrastructures [1, 2, 3]. Improvements in the performance can result in a drop in overall computational and financial costs. This optimization can be the result of the analysis of the motorization system: contributions such as [4] describe the typical complexities and uncertainties in this kind of infrastructures, like resource contention and its uncertainty or robustness, to address real-time problems through robust big data solutions.

Recent advances in Remote Sensing technology have improved the volume of the available image data. In this contribution, we consider the data available in the Open Data Euskadi repository [1]. The ortho-imagery, acquired annually by an air-borne camera, is characterized by a spatial resolution of 25 cm in each direction. As a consequence of technological limitations in the acquisition and storage systems, the number of available channels is limited to three. The available coverage spans to the whole extension of the Basque Country, resulting in around 1500 GeoTIFF products with a size of about 33Gb each, as per figure 3. The need for an effective divide and conquer approach based on data tiling and parallel processing is evident in this context.

2. SYSTEM DESCRIPTION

The prototype presented in [5] is a web map server with integrated machine learning — including classification and clustering — capabilities, able to create thematic coverage maps. The prototype is if needed able to distribute processing loads by big data frameworks like Apache Spark [6], taking full advantage of distributed in-memory computing.

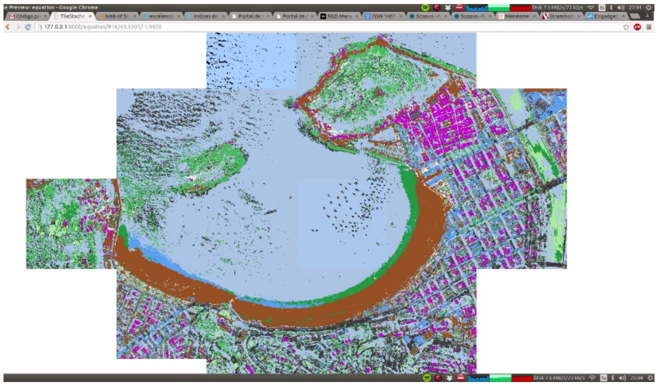

Fig. 1. Prototype UI for real time execution of supervised classification tasks across the map. The thematic coverage tiling process is similar to typical tiled map navigation data service, yet it includes components that allow causal users to define thematic classes of interest that might be unique to their interests and activities.

The operation of the prototype for supervised classification can be described as follows. A trained classification model is distributed to each worker operating in the infrastructure for the generation of the thematic map. While interactively navigating the map, each tile of the thematic coverage map is created by a worker node in a lazy mode, based on the HTTP requests put forward by an HTML-based Graphical User Interface implementing a standard web map navigator. As can be see in figure 1, the thematic coverage tiling process is similar to typical tiled map navigation interfaces, yet it includes components that allow causal users to define thematic

[1] `http://opendata.euskadi.net/`

classes of interest that might be unique to their interests and activities. To obtain each tile, different features are computed according to the training, in order to process them by the distributed trained classification model.

The architecture of the server side (figure 2) starts conceptually with data archives in charge of storing the original data. The data can in turn be analyzed by a scalable set of processors co-located with the archive. External users can access either the original or the processed data, as well as exploit higher-level descriptions of data content also available as Services defined by the system, by accessing application servers dedicated to specific data exploitation scenarios such as those dedicated to specific applications in forestry or in the exploitation of marine resources.

Computing resource consumption measures like CPU and memory need to be monitored closely in order to analyze the behavior of the system in the face of varying user demand. Further measures corresponding to the inner operations of the training and classification system can be very useful to obtain intermediate information or to improve the performance of the prototype.

The monitoring of these metrics is the focus of the present work.

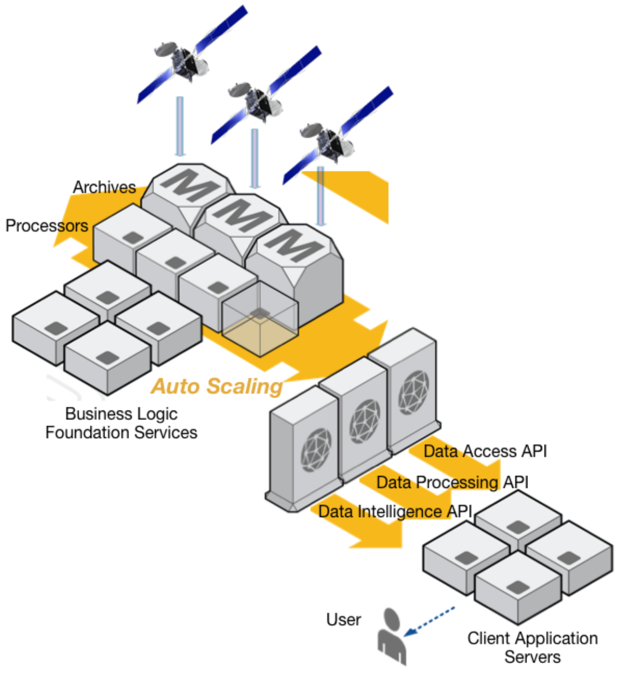

Fig. 2. System architecture. Archives in charge of storing the original data are served by a scalable set of processors co-located with the archive. External users can access either the original or the processed data, as well as exploit higher-level descriptions of data content also available as Services defined by the system.

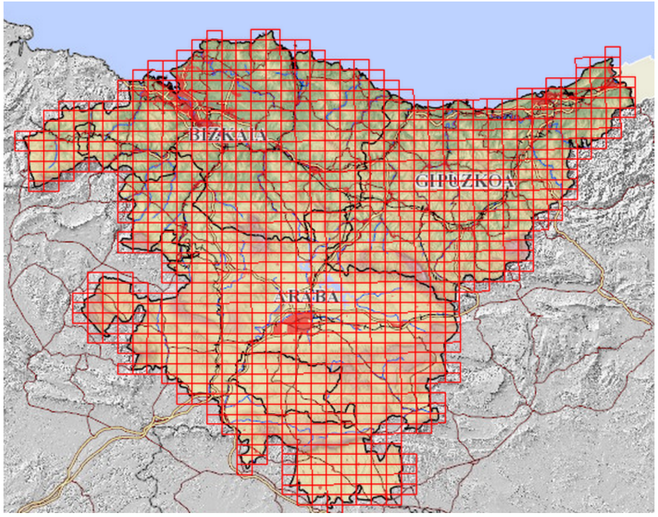

Fig. 3. Available GeoTIFF product map. The ortho-imagery, acquired annually by an air-borne camera, is characterized by a spatial resolution of 25 cm in each direction. As a consequence of technological limitations in the acquisition and storage systems, the number of available channels is limited to three. The available coverage spans to the whole extension of the Basque Country, resulting in around 1500 GeoTIFF products with a size of about 33Gb each.

2.1. Monitoring subsystem

The proposed monitoring system tries to optimize the distributed thematic mapping process by minimizing the impact on the system performance: because the user interface of this prototype is web based, processing time acquires a unusual relevance in machine learning processes where time limitations usually play a less prominent role [7].

The main task of the server is to create thematic coverage tiles based on common machine learning models, to build a coherent coverage map.

System quality as related to the classification results measured with metrics like Precision, Recall and F1 are not the object of the current contribution: they have been extensively described in [5] and [8].

The monitoring of the computing performance of the different modules in the system is here understood as an internal description of the operation of the system, rather than being related to external ground truth maps. From an architectural point of view, the analyzed performance metrics are recollected in each worker node and aggregated in a specialized node of the infrastructure destined to this task. Using a specialized web based user interface dedicated to the administration of the system, these values can be monitored in real time as per Figure 4.

Whenever required, these metrics could be represented and analyzed statistically using multi-variate analysis and representation methods, to analyze the behavior of the system

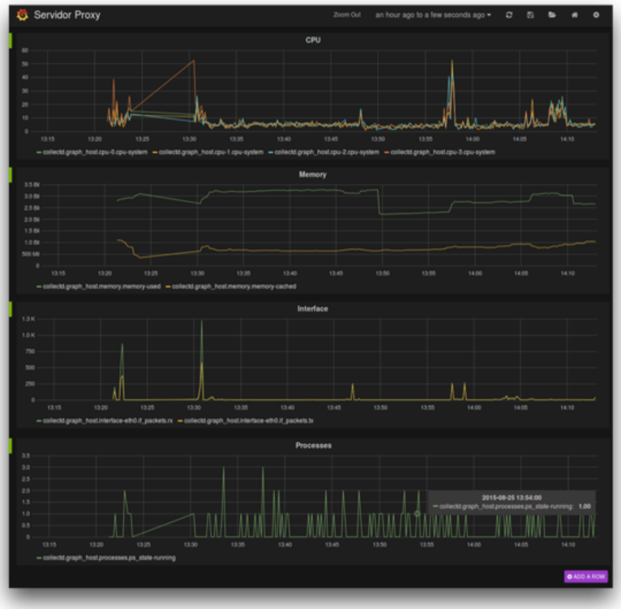
Servidor Proxy
Zoom Out
an hour ago to a few seconds ago
CPU
60
50
40
30
20
10
0
13:15
13:20
13:25
13:30
13:35
13:40
13:45
13:50
13:55
14:00
14:05
14:10
collectd.graph_host.cpu-0.cpu-system
collectd.graph_host.cpu-1.cpu-system
collectd.graph_host.cpu-2.cpu-system
collectd.graph_host.cpu-3.cpu-system
Memory
3.5 Bi
3.0 Bi
2.5 Bi
2.0 Bi
1.5 Bi
1.0 Bi
500 Mi
0
collectd.graph_host.memory.memory-used
collectd.graph_host.memory.memory-cached
Interface
1.3 K
1.0 K
750
500
250
0
collectd.graph_host.interface-eth0.if_packets.rx
collectd.graph_host.interface-eth0.if_packets.tx
Processes
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
2015-08-25 13:54:00
collectd.graph_host.processes.ps_state-running: 1.00
collectd.graph_host.processes.ps_state-running
ADD A ROW

Fig. 4. Real time monitoring system UI. The UI allows to follow the operation of the prototype in real time, contributing to its understanding and allowing operators to optimize its processing mechanisms.

and to try and determine the most appropriate actions that allow the improvement of the performance of the system in any of the presented contexts.

3. MONITORING RESULTS

As stated, the generated metrics are collected to serve the internal optimization of the system.

Yet, when the system is applied to a specific dataset, their content is naturally related to the characteristics of this data.

A specific case is reported in relation to figure 5.

The model learning time is in this case reported for each visited tile across a scene covering the city of Donostia/San Sebastián. The considered machine learning model is an instance of the K-Means algorithm [9]. For investigating the observable variations if the classification costs in the prototype, the model is re-trained separately on each one of the coverage tiles — naturally resulting in incompatible assignments across the whole coverage. The measured processing times are represented in terms of a heatmap-like layer that we superimpose to a map of the processed area, as in figure 5. Variations in processing time seem to be explicable in terms of the variability of the input data: very flat areas result in slower analysis operations (a result that can easily be replicated in the case of many k-means implementations), or perhaps natural areas characterized by more inherent complexity generate data that has more variation to be accounted for in the analysis.

4. CONCLUSIONS

Optimizing the exploitation of big data infrastructures for distributed thematic mapping at global scales requires an optimal use of the resources to reduce computational and financial costs.

A monitoring system allows to control these infrastructures so that they can attain better performance by applying appropriate measures, allowing operators to develop own metrics in addition to the classics that enrich the knowledge of the system.

This understanding of system performance effects can leverage the exploitation of machine learning methodologies able for instance to identify the most important metrics or to synthesize summary measures with increased information content.

This better knowledge of the inner workings of the system can produce a better performance of the monitored system and therefore the capability of analyzing much more extended coverage maps with reduced computational costs.

5. REFERENCES

[1] Dan Wang and Jiangchuan Liu, “Optimizing big data processing performance in the public cloud: opportunities and approaches,” *Network, IEEE*, vol. 29, no. 5, pp. 31–35, September 2015.

[2] Stefano Nativi, Paolo Mazzetti, Mattia Santoro, Fabrizio Papeschi, Max Craglia, and Osamu Ochiai, “Big data challenges in building the global earth observation system of systems,” *Environmental Modelling & Software*, vol. 68, pp. 1–26, 2015.

[3] Yan Ma, Haiping Wu, Lizhe Wang, Bormin Huang, Rajiv Ranjan, Albert Zomaya, and Wei Jie, “Remote sensing big data computing: Challenges and opportunities,” *Future Generation Computer Systems*, vol. 51, pp. 47–60, 2015.

[4] R. Ranjan, “Modeling and simulation in performance optimization of big data processing frameworks,” *Cloud Computing, IEEE*, vol. 1, no. 4, pp. 14–19, Nov 2014.

[5] J. Lozano, N. Aginako, M. Quartulli, I.G. Olaizola, E. Zulueta, and P. Iriondo, “Large scale thematic mapping by supervised machine learning on ‘big data’ distributed cluster computing frameworks,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, July 2015, pp. 1504–1507.

[6] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica, “Spark: Cluster computing with working sets,” in *Proceedings of the*

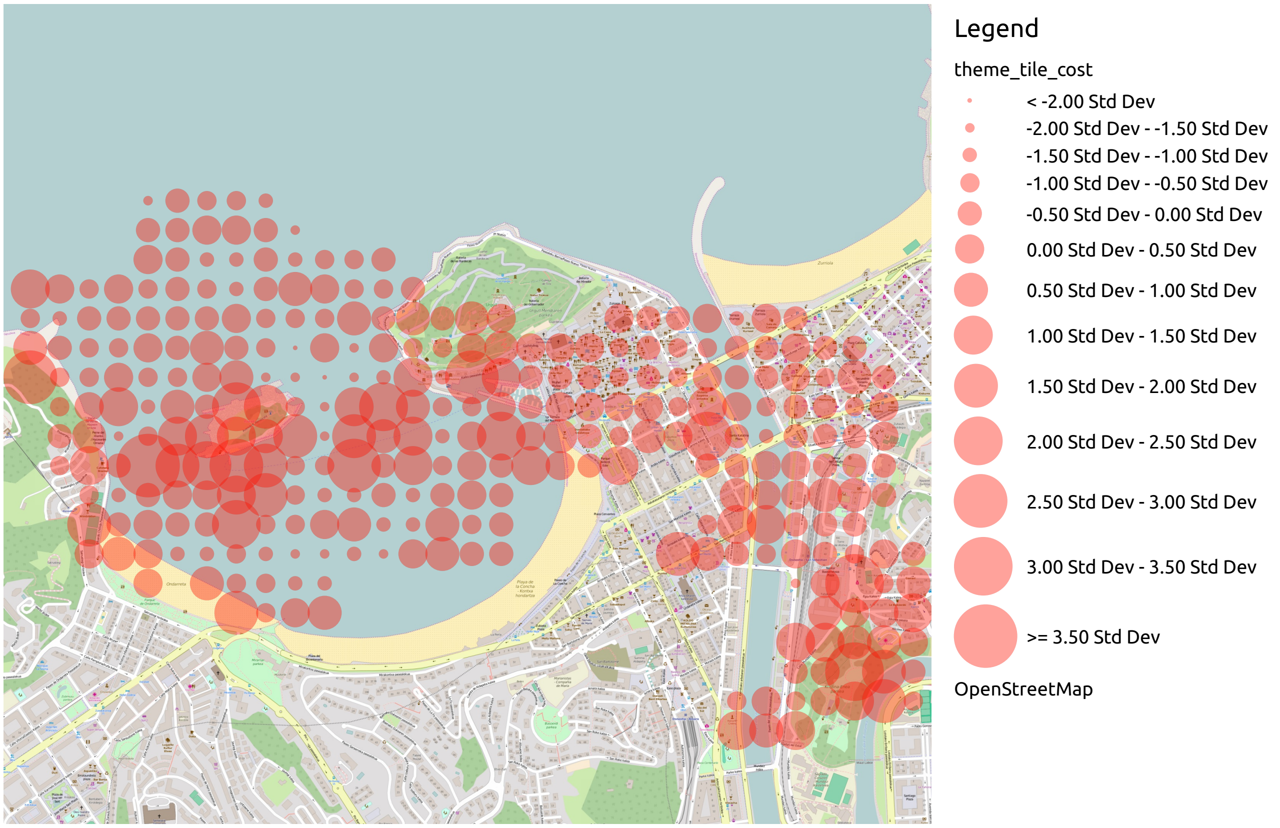
Legend
theme_tile_cost
< -2.00 Std Dev
-2.00 Std Dev - -1.50 Std Dev
-1.50 Std Dev - -1.00 Std Dev
-1.00 Std Dev - -0.50 Std Dev
-0.50 Std Dev - 0.00 Std Dev
0.00 Std Dev - 0.50 Std Dev
0.50 Std Dev - 1.00 Std Dev
1.00 Std Dev - 1.50 Std Dev
1.50 Std Dev - 2.00 Std Dev
2.00 Std Dev - 2.50 Std Dev
2.50 Std Dev - 3.00 Std Dev
3.00 Std Dev - 3.50 Std Dev
>= 3.50 Std Dev
OpenStreetMap

Fig. 5. K-Means clustering model learning time is represented for each visited tile across a scene covering the city of Donostia/San Sebastián. For investigating the observable variations if the classification costs in the prototype, the model is re-trained separately on each one of the coverage tiles — naturally resulting in incompatible assignments across the whole coverage. As is typical for k-means implementations, variations in processing time seem to be explicable in terms of the variability of the input data: very flat areas result in slower analysis operations (a result that can easily be replicated in the case of many k-means implementations), or perhaps natural areas characterized by more inherent complexity generate data that has more variation to be accounted for in the analysis.

2Nd USENIX Conference on Hot Topics in Cloud Computing, Berkeley, CA, USA, 2010, HotCloud'10, pp. 10–10, USENIX Association.

[7] Fiona Fui-Hoon Nah, "A study on tolerable waiting time: how long are web users willing to wait?," *BEHAVIOUR & INFORMATION TECHNOLOGY*, vol. 23, no. 3, pp. 153–163, 2004.

[8] J. Lozano Silva, N. Aginako Bengoa, M. Quartulli, I.G. Olaizola, and E. Zulueta, "Web-based supervised thematic mapping," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 8, no. 5, pp. 2165–2176, May 2015.

[9] Richard O Duda, Peter E Hart, and David G Stork, *Pattern classification*, John Wiley & Sons, 2012.