# BEYOND THE LAMBDA ARCHITECTURE: EFFECTIVE SCHEDULING FOR LARGE SCALE EO INFORMATION MINING AND INTERACTIVE THEMATIC MAPPING

*Marco Quartulli, Javier Lozano, Igor G. Olaizola*

Vicomtech-IK4, Mikeletegi 57, 20009 Donostia, Spain
E-mail: {mquartulli, jlozano, iolaizola}@vicomtech.org

## 1. INTRODUCTION

As per the 2013 EOSDIS annual metrics report[1], Petabyte-scale Earth Observation (EO) raster data archive volumes are growing at rates of about ten Gigabytes per day while around 95% of their contents have never been accessed by a human observer [1]. Metadata-based search clearly needs to be complemented by semi-automatic raster catalog content mining.

While large scale near real-time data processing is a crucial component of EO mining for the interactive semantic labeling of archive contents, available architectures [2] typically only permit batch processing strategies on large scale coverages, hence trying to adapt continuous ingestion from rolling archives and catalog content index updates to this prevalent paradigm.

In this contribution, we propose instead to consider the streaming approach as the standard one for EO mining, adapting large and historical batch data processing to a near-real time scenario by task queueing and smart scheduling. To this end, we show how the organization of the data on N-dimensional lattices and the locality of access patterns in the spatial and in the resolution dimensions allow us to define and exploit a methodology that is based on streaming cluster computing frameworks [3], Hilbert curve scheduling [4] and multi-scale pyramid decompositions for optimizing access to distributed storage and computing resources and maximize perceived processing speed in interactive operations.

In doing so, we propose a methodology to improve on the 'Lambda architecture' [5], the prevalent approach in managing the contradiction between the large sizes of remote sensing products and the significant data rates their processing involves, especially in the interactive training sessions needed for semantic labeling of the archive contents. The Lambda architecture solves the problem of computing arbitrary functions on arbitrary data in realtime by combining (as in a $\Lambda$-shaped diagram) a batch layer for processing large scale historical data and a streaming layer for processing items being retrieved in real time from an input queue. While resulting systems have good scalability characteristics, their composition in terms of two different and typically incompatible processing paradigms represents a significant disadvantage in terms of programming complexity and maintainability. Our approach effectively supersedes the Lambda architecture for one focusing on streaming exclusively for both the near-real time and the batch processing in the design of distributed EO mining systems.

The specific and simplified scenario we consider is that of interactive thematic mapping over Gigapixel input coverages supervised via a tile-based user interface (UI). As per [6], machine learning algorithms based on cluster computing frameworks [7] can be used to try and manage the rapid growth in processing costs, yet in the case of near-real time processing their operation can effectively be improved by the adoption of specific scheduling mechanisms.

## 2. METHODS

Assuming the input coverage is appropriately decomposed into a multi-resolution pyramidal tiling, a divide-and-conquer approach can be used to distribute the processing to multiple worker units. Most approaches, including the prevalent MapReduce model [8], treat the resulting concurrent process-

[1] `https://earthdata.nasa.gov/sites/default/files/field/document/FY13AnnualReport_V4.xlsx`, visited 09 Jan 2015

ing lines as equivalent in priority. This means that in the perfectly parallel large scale supervised machine learning scenario we consider, the required processing time for the tile of interest $t_i$ is of the order $O(O_1 \cdot N/W)$ with $N$ the number of total tiles, $W$ the number of workers allocated to the problem, and $O_1$ the worst-case asymptotic complexity of the classification of a single tile.

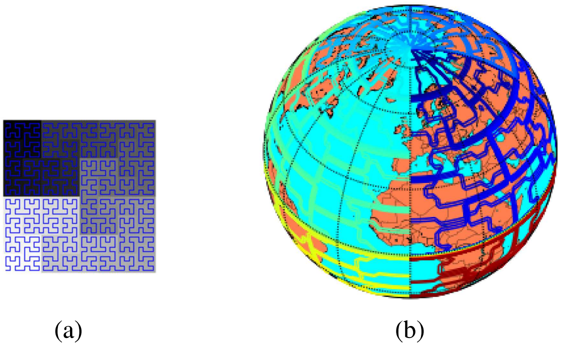

**Fig. 1**. 2D Hilbert curves with associated partitioning (a) on the unit square and (b) on a global geographic coverage (b).

An alternative can be found in the adoption of effective scheduling mechanisms. A linear order for sorting and scheduling objects that lie in a multi-dimensional space can be imposed by fractal space-filling curves [9]. Hilbert curves in particular have been proposed as coloring mechanisms for high-dimensional spaces in visualization [10] and for declustering in distributed multimedia search [11], and are used for assigning related tasks to locations with higher levels of proximity in parallel processing systems [4]. We propose to extend a scheduling strategy based on Hilbert curves to large scale processing of raster data defined on geographic spaces augmented by the resolution dimension. This scheme naturally allows considering computed and pre-existing relevance maps by composing the priority for a given tile with raster measures defined on the same lattice as the tiles, e.g. for filtering out large extensions of irrelevant land cover. The required processing time for the tile of interest $t_i$ is in this case of the order $O(O_1 \cdot C_i/W)$ with $C_i$ the ranking of the tile of interest on the curve considered by the scheduler, which is stricly smaller than $N/W$. A scheduler capable of exploiting the priority measures hitherto defined can be built on top of streaming cluster computing frameworks to consume events generated by a queue loaded by traversing a path defined by the space-filling curve on geographic relevance maps. This effectively entails a fusion of the two branches of the Lambda architecture into a single stream-oriented design incorporating smart task scheduling mechanisms in the loading of the input queue.

A parallel can be drawn in between standard MapReduce and the proposed method on one side and the concepts of systematic and stratified sampling in statistics.

Standard data analytics relies on sampling and inferential statistics for testing the validity of hypotheses on the properties of large populations. A basic example of a sampling methodology is represented by systematic sampling, in which an ordered or pre-arranged set of data points is iterated upon by selecting elements at regular intervals. An alternative is stratified sampling [12], in which the population is segmented into mutually exclusive sub-groups that are subsequently sampled independently. The stratification tends to be effective when the criterion is strongly correlated with the dependent variable being investigated.

'Big Data' methodologies focus on descriptive rather than inferential statistics, and on analyzing quasi-complete data closely matching a population instead of sampling. To do that, the data points are split into mutually exclusive groups by simple methodologies resembling systematic sampling. The statistics computed separately for the different groups are typically combined in a 'Reduce' step.

The methodology we propose relies aims at improving the performance of a data analysis system by executing in parallel the analysis of a specific stratum of interest before extending the computation to sections of the data volume further from the one of interest, effectively exploiting a continuous version of the criterion used for stratified sampling as an indicator of precedence that is considered among the inputs of a parallel processing scheduling system.

## 3. RESULTS

A small prototype-level implementation of the proposed concept can be described in terms of a series of simple components as in the following paragraphs. A multi-resolution pyramidal tile decomposition is generated from a Virtual Raster dataset fusing multiple products. In the perfectly parallel batch analysis architecture typically employed in distributed EO mining systems of the type described in [6] (fig. 2a), the generated tiles are saved onto a distributed filesystem whose

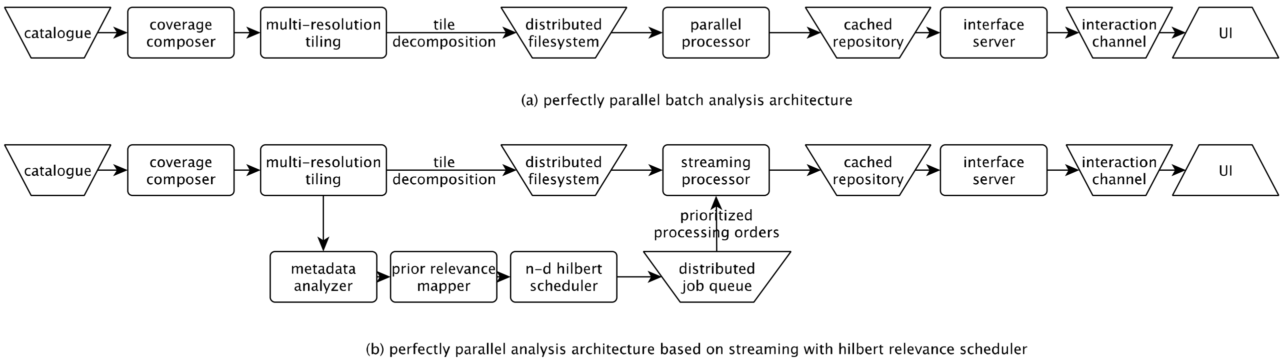

**Fig. 2**. High-level architectural description: a perfectly parallel analysis architecture corresponding to the batch data branch of the Lambda architecture employed in distributed EO mining systems of the type of [6], in which all tiles are processed with the same priority, can effectively be supplemented by an architecture based on streaming centered on a Hilbert scheduling mechanism.

contents feed a queue that is processed in parallel, serving the results in a cache repository to a user interface via an interaction channel. In our prototype based on the described approach (fig. 2b), metadata for the resulting tiled space are instead exploited by an N-dimensional Hilbert curve generator that is traversed by a scheduler defining the priority levels for the tasks inserted in a distributed FIFO queue. The elements in the queue are consumed by a streaming map-reduce client that outputs the obtained results into an interaction channel that in turn feeds the supervision user interface via a web socket.

We run a minimal validation experiment on a $16 \times 16$ tiles decomposition of an input dataset with 8 worker nodes, in which processing times for a single tile amount to about 1.4 seconds. Results are reported in figure 3 in terms of the time needed for the completion of the processing on the tiles near to the area in the pyramid being examined by the human analyst. While the exploitation of parallel computing reduces the cost of about an order of magnitude with respect to the serial processing case, the exploitation of a scheduling methodology in the proposed architecture reduces the processing cost of about two and of more than three orders of magnitude with respect to the serial and the parallel processing scenarios respectively.

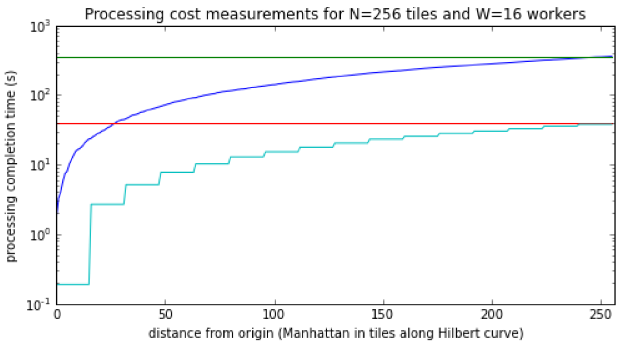

**Fig. 3**. Experimental results in terms of the time needed for the completion of the processing on the tiles near to the area in the pyramid being examined by the human analyst. The green line corresponds to a single processing worker and serial unscheduled processing. The blue line to a single processor and a serial Hilbert-scheduled stream. The red line to a multiprocessor with simple parallel processing. The cyan line to a multiprocessor with a Hilbert-scheduled processing stream. Typical processing times for a single tile are around 1.4 seconds. While the exploitation of parallel computing reduces the cost of about an order of magnitude with respect to the serial processing case, the exploitation of a scheduling methodology in the proposed architecture reduces the processing cost of about two and of more than three orders of magnitude with respect to the serial and the parallel processing scenarios respectively.

## 4. CONCLUSIONS

The present contributions introduces and demonstrates a methodology that can be used to improve on standard perfectly parallel processing architectures for near real-time interactive thematic mapping and EO catalog mining on very large scale coverages.

It has to be noted that the significance of the defined priority assignment mechanism is not strictly limited to parallel architectures running on cluster processing frameworks. It can be employed even on single CPU machines for prioritizing the processing of raster tiles in interactive analysis scenarios.

## 5. REFERENCES

[1] Manolis Koubarakis, Michael Sioutis, George Garbis, Manos Karpathiotakis, Kostis Kyzirakos, Charalampos Nikolaou, Konstantina Bereta, Stavros Vassos, Corneliu Octavian Dumitru, Daniela Espinoza-Molina, et al., "Building virtual earth observatories using ontologies, linked geospatial data and knowledge discovery algorithms," in *On the Move to Meaningful Internet Systems: OTM 2012*, pp. 932–949. Springer, 2012.

[2] Marco Quartulli and Igor G Olaizola, "A review of EO image information mining," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 75, pp. 11–28, 2013.

[3] Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, and Ion Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*. USENIX Association, 2012, pp. 10–10.

[4] Maciej Drozdowski, *Scheduling for parallel processing*, Springer, 2010.

[5] Nathan Marz and James Warren, *Big Data: Principles and best practices of scalable realtime data systems*, O'Reilly Media, 2013.

[6] Luigi Mascolo, Marco Quartulli, Pietro Guccione, Giovanni Nico, and Igor G. Olaizola, "Scalable analysis and retrieval of SAR data on Elastic Computing Clouds," in *Proceedings of the 2014 conference on Big Data from Space (BiDS'14)* . IEEE, 2014, pp. 150–153.

[7] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica, "Resilient distributed datasets: a fault-tolerant abstraction for inmemory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012, pp. 2–2.

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

[9] Mir Ashfaque Ali and SA Ladhake, "Overview of space-filling curves and their applications in scheduling," *International Journal of Advances in Engineering and Technology*, vol. 1, no. 4, pp. 148–154, 2011.

[10] Barry Irwin and Nick Pilkington, "High level internet scale traffic visualization using hilbert curve mapping," in *VizSEC 2007*, pp. 147–158. Springer, 2008.

[11] Stefan Berchtold, Christian Böhm, Bernhard Braunmüller, Daniel A Keim, and Hans-Peter Kriegel, "Fast parallel similarity search in multimedia databases," in *Proc. ACM SIGMOD Int. Conf. on Management of Data*. 1997, vol. 26 of *2*, ACM.

[12] Mohammad Shahrokh Esfahani and Edward R Dougherty, "Effect of separate sampling on classification accuracy," *Bioinformatics*, p. btt662, 2013.