The δ big data architecture for mobility analytics

George Vouros, Apostolis Glenis and Christos Doulkeridis

Abstract Motivated by needs in mobility analytics that require joint exploitation of streamed and voluminous archival data from diverse and heterogeneous data sources, this chapter presents the δ architecture: Denoting “difference", δ emphasises on the different processing requirements from loosely-coupled components, which serve intertwined processing purposes, forming processing pipelines. The δ architecture, being a generic architectural paradigm for realizing big data analytics systems, contributes principles for realizing such systems, focusing on the requirements from the system as whole, as well as from individual components and pipelines. The chapter presents the datAcron integrated system as a specific instantiation of the δ architecture, aiming to satisfy requirements for big data mobility analytics, exploiting real-world mobility data for performing realtime and batch analysis tasks.

1 Introduction

The technical challenges associated with Big Data analysis are manifold, and perhaps better illustrated in [3] using a Big Data Analysis Pipeline, such as the one depicted in Figure 1. Similar pipelines have been proposed elsewhere, e.g. the Ingest-Enrich-Store-Train-Query pipeline of IBM Watson discovery generic process.

George Vouros
University of Piraeus, Greece
e-mail: georgev@unipi.gr

Apostolis Glenis
University of Piraeus, Greece
e-mail: aglenis@unipi.gr

Christos Doulkeridis
University of Piraeus, Greece
e-mail: cdoulk@unipi.gr

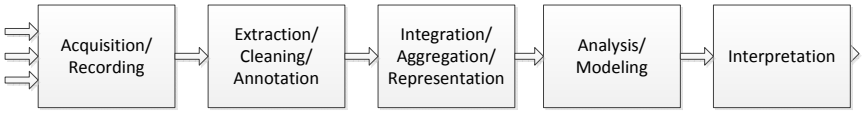

Fig. 1: The big data analysis process.

The different snapshots of the big data analysis process result from the need to "ingest and digest", i.e. gather, process, integrate and analyse the increasing amounts of data coming from different data sources. These sources may include sensors providing data-in-motion (streamed data), or stores providing data-at-rest (archival or historical data). The aim is to satisfy requirements for realizing internet of things and cyber-physical systems exploiting big data, as well as for advancing the human monitoring, awareness, prediction and decision making abilities in critical domains, such as in transportation and traffic. As discussed in the first two chapters of this book, the goal is to support data analytics in order to reveal models that provide sufficient abilities towards identifying and predicting important happenings, occurring trends, important situations, as well as prescribing appropriate actions and plans.

The major question to be answered is *how such an abstract process should be realized as a big data architecture*, orchestrating the functionality and satisfying domain specific requirements related (a) to the variety of data coming from different, isolated data sources designed for different purposes, and including data with different semantics and formats; (b) to the volume, velocity and veracity of data to be processed and/or managed by the overall system.

Proposals for paradigmatic architectures or architectural patterns for big data systems emphasizing on "analytics" include the λ and κ architectures. The λ architecture separates the batch from the realtime processing needs, incorporating components that realize the same data processing tasks in separate layers. However, to resolve a query function [2], one has to merge results from the batch view and the realtime view. This blending of results from different layers may hinder the satisfaction of realtime requirements. Aiming to proving results from a realtime layer without using a batch layer, the κ architecture [1] considers that any data source can be treated as a provider of streamed data. This view impacts the way we do batch processing: For instance, we may need to fetch a well-selected subset of pre-processed historical data from a store, not retained in any log, and do any batch task that inherently requires voluminous data, computationally demanding processing steps, human involvement, application of iterative exploration-filtering-subset selection, data-transformation steps in visual analytics workflows, etc. Thus, we need a type of architecture where both batch and realtime layers co-exist, following the λ prescriptions, but enabling different functionality at different layers of processing, allowing components to run quite independently from those in other layer(s), much like the κ paradigm. Hence, to realize the overall big data pipeline, we need to decide on the arrangement of individual components in architecture layers according to functionality and performance requirements. This may result to multiple pipelines, each

one serving specific purposes and adhering to specific performance requirements. Different pipelines may share components or incorporate components that realize the same data processing tasks, much like the λ paradigm.

The work reported in this chapter is motivated by requirements on mobility analytics in time-critical domains, specifically, in aviation and maritime (as these have been presented in the first part of this book), and contributes a paradigmatic big data architecture that emphasises on the need for different layers of processing, targeting to different, albeit interacting, realtime and batch analytics tasks, aiming to satisfy their performance requirements. We opt for denoting the differences on layers' components requirements, and call the architecture "delta", i.e. δ.

The δ architecture, going beyond mobility analytics tasks, contributes generic principles for incorporating components into a layered architecture, realizing multiple facets of the generic big data analysis pipeline, where each component may function both, as a consumer and as a producer. In doing so, we clearly separate the functionality and performance requirements from each of the components and provide rules on performance constraints that should be satisfied by pipelines. The article describes an implemented instantiation of the δ architecture: The datAcron integrated system for real-world big mobility data analytics.

This chapter presents background concepts and requirements for exploiting mobility data in time critical domains and presents the proposed δ architecture paradigm and snapshots of the δ architecture. Then, it presents a specific realization of δ: The datAcron implemented prototype for mobility analytics, explaining specific architectural choices to support performance requirements in time critical domains.

References

1. J. Kreps, “Questioning the Lambda Architecture". radar.oreilly.com. O’reilly, July 2, 2014. Retrieved 10 May 2018.
2. N. Marz, “How to beat the CAP theorem”, nathanmartz.com/blog, October 13, 2011. Retrieved 10 May 2018.
3. H. V. Jagadish, et al., “Big data and its technical challenges”, Commun. ACM, 57(7):86–94, 2014.