Using Big Data in Healthcare

E

Georgios Lamprinakos
National Technical University of Athens, Greece

Ioanna A. Aristeidopoulou
Intracom Telecom, Greece

Stefan Asanin
Anticimex, Greece

Andreas P. Kapsalis
National Technical University of Athens, Greece

Angelos-Christos G. Anadiotis
National Technical University of Athens, Greece

Dimitra I. Kaklamani
National Technical University of Athens, Greece

Iakovos S. Venieris
National Technical University of Athens, Greece

INTRODUCTION

The healthcare sector is undergoing fundamental changes recently. The vision of transforming the healthcare system from a treatment-oriented system to a patient-centered system based on prevention, ubiquitous monitoring and continuous multi-level support is considered to be a key point towards the improvement of the provided services and the reduction of healthcare costs, mainly attributed to the expected reduction in the number of hospitalizations. Throughout the new pathways, data volume and complexity are increasingly augmenting and require the use of the emerging big data analytics technologies in order to be significantly exploited. Health and social care services integration, continuity of care to the home, behavioral analysis and remote monitoring based on sensing devices are just a few elements of the new generation healthcare domain that all lead to the management of large, complex and frequently heterogeneous data.

In this chapter, the authors will summarize the trends of the modern healthcare ecosystem and present the current application categories, as well as challenges posed, concerning the high volume, variety and velocity healthcare data manipulation.

BACKGROUND

The term ‘big data’ refers to high volume and complexity data, which have become extremely difficult to process using traditional techniques. One of the earliest definitions of big data was provided by Laney

DOI: 10.4018/978-1-4666-9978-6.ch068

(2001), describing the new data growth challenges as three-dimensional (*Volume, Velocity, Variety*). The three-dimensional model, which stands as the most adopted definition up to date, regards big data as data that are massive in size (*Volume*), need to be quickly acquired and processed (*Velocity*), and are unstructured, loosely coupled and cannot be handled by traditional RDBMS (*Variety*). Another 'V' term that frequently accompanies the definition of big data is *Veracity*, indicating the need of processing clean and quality data, in order to achieve meaningful results from the big data use (cf. e.g., Buhl et al., 2013).

Nowadays, various sectors have transformed their way of functioning and they are collecting and exploiting large amounts of data, that may offer groundbreaking insights. Traditional services, such as weather forecasting, now require the manipulation of large amounts of data in order to provide better models, more accurate predictions, profiling and decision support. Many other paradigms exist where big data analytics can reshape well established operations. Power grids, for example, are beginning to evolve into smart grids, which are able to analyze data acquired from smart home appliances and smart sensors in order to sustain resources and energy and prevent service downtime during peak hours (Balac et al., 2013).

The aforementioned achievements, would not be made possible using legacy software methods and database systems. During the last years, extensive research has been conducted in order to invent new programming paradigms and software that would improve the manipulation, processing and analyzing of large scale data. MapReduce is a prevailing programming model for processing large datasets (Dean & Ghemawat, 2008), which usually runs on a cluster by using a parallel/distributed algorithm. Several MapReduce implementations exist, with Apache Hadoop (ASF, 2014a) being the most popular. The Hadoop framework is composed by several modules that include file-system for distributed storage, resource-management services and data processing module which is of course an implementation of the MapReduce algorithm. Among other MapReduce implementations are the CouchDB open source database system (ASF, 2014b), Infinispan (Red Hat, 2014) and MongoDB (MongoDB, 2014). Another fundamental big data technical component is Bigtable (Chang et al., 2006), which proposes a distributed file storage system for managing data that can scale to a very large size. Bigtable uses the distributed Google File System to store logs and data files. Some well-known open-source implementations of Bigtable, are Accumulo (ASF, 2014c), Cassandra (ASF, 2014d) and HBase (ASF, 2014e), all offered by the Apache Software Foundation. The successors of Bigtable are designed to meet the requirements of today's interactive online services. Google's Megastore provides fully serializable ACID semantics within fine-grained partitions of data (Baker et al., 2011), while Spanner is Google's scalable, multi-version, globally-distributed and synchronously-replicated database (Corbett et al., 2013). Other related software projects include Dremel, which is a query system for data analysis, presented by Melnik et al. (2010), Pregel, which is a scalable and fault tolerant graph processing system developed by Malewicz et al. (2010) and Tenzing, which offers an SQL implementation built on top of MapReduce for querying large datasets (Chattopadhyay et al., 2011).

The operations that are required for processing large datasets, as the ones present in the big data era, are computationally intensive and could take an unacceptable amount of time to yield the desired results. Facing this challenge, cloud computing has emerged as an optimal solution for executing jobs and tasks of that magnitude (Armbrust et al., 2010). The cloud computing model offers a collection of heterogeneous computing, storage and network resources as a service. Three basic types of cloud services are offered: IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service) and SaaS (Software-as-a-Service). Computationally intensive tasks, such as MapReduce jobs, can benefit from the elasticity, high availability, fault-tolerance and transparency features of the Cloud. In fact, many MapReduce implementations are supported and offered out-of-the-box by many Cloud software providers, such as Apache CloudStack