Big Data Analytics Technologies and Platforms: a *brief* review

Ticiana L. Coelho da Silva[1], Regis P. Magalhães[1], Igo R. Brilhante[1], José Antonio F. de Macêdo[1], David Araújo[1], Paulo A. L. Rego[1], Aloisio Vieira Lira Neto[2]

[1]Federal University of Ceará, Brazil

[2]Brazilian Federal Highway Police

{ticianalc, regismagalhaes, pauloalr}@ufc.br, aloisio.lira@prf.gov.br

{igobrilhante, jose.macedo, araujodavid}@lia.ufc.br

Abstract. *A plethora of Big Data Analytics technologies and platforms have been proposed in the last years. However, in 2017, only 53% of companies are adopting such tools. It seems that the industry is not convinced about Big Data promises or maybe choosing the right technology/platform requires in-depth knowledge about the capabilities of all these tools. Before deciding the right technology or platform to choose from, the organizations have to investigate the application/algorithm needs and the advantages and drawbacks of each technology/platform. In this paper, we aim at helping organizations in the selection of technologies/platforms more appropriate to their analytic processes by offering a short-review according to some categories of Big Data problems as processing (streaming and batch), storage, data integration, analytics, data governance, and monitoring.*

1. Introduction

According to [Dresner Advisory Services 2017], 53% of companies were adopting Big Data Analytics platforms. Such news and other similar ones motivate a reflection on why Big Data adoption is not yet a reality to many companies. On the one hand, we are witnessing a data deluge, which demands scalable solutions to extract value from a large volume of data. On the other hand, Big Data technologies are usually presented as the key answer to such need. However, it seems that the industry is not convinced about Big Data promises. To our understanding this problem comes from the lack of a thorough knowledge of what is a *big data problem* and what are the advantages and drawbacks of the available *big data technologies*.

The management and analysis of large-scale datasets are usually associated with the term Big Data. According to [Begoli and Horey 2012], Big Data is the practice of collection and processing of large datasets, systems, and algorithms used to analyze these massive datasets. [Wu et al. 2014] claims that Big Data refers to large heterogeneous volumes of autonomous sources with distributed and decentralized control, trying to explore the complex and evolving relationships between data. Because of the distributed processing involving lots of nodes, it is necessary that the data management in Big Data deals with a failure of the nodes as a frequent event, and not as an exception to the processing. Meanwhile, Gartner introduced Big Data as characterized by three Vs: volume, variety, and velocity [Sakr 2016]. After the three Vs definition was extended to four Vs, with the addition of value.

In this paper, we analyze Big Data from two different perspectives: Big Data technologies and Big Data platforms. Several research studies [Cohen et al. 2009, Herodotou et al. 2011, Begoli and Horey 2012, Singh and Reddy 2015], together with the authors' development experience with Big Data problems, advocate that all Big Data technologies should require fault tolerance, scalability, elasticity, distributed architecture, generic storage and processing of large volumes of data in the order of terabytes or even petabytes. Besides, a Big Data platform with an ecosystem of services and technologies should also provide resource management, data governance, and monitoring. In this work, we only refer to the technologies and platforms regarding such features.

Notably, we aim at comparing different Big Data technologies and analytics platforms according to the following categories: processing (streaming and batch), storage, data integration, analytics, governance, and monitoring. There exist several papers that compare big data technologies, to name a few [Inoubli et al. 2018, Sakr 2016, Singh and Reddy 2015]. However, they do not address big data platform analytics. On the other hand, this study aims to help organizations in the selection of platforms more suitable to their analytic processes. Since, typically, before deciding on the right technology or platform to choose from, the user/organization investigates what the application/algorithm needs are and what each technology/platform may provide. It is worth to mention that our focus is not comparing Big Data technologies and platforms for different applications, like Cloud Computing and Internet of Things, but to compare them according to categories of Big Data problems.

The remainder of the paper is structured as follows: Section 2 and 3 provide an overview of the relevant Big Data technologies and platforms, respectively, from the state-of-the-art works. Moreover, these sections present a comparison of such technologies and platforms based on some categories of problems. Finally, Section 4 draws final considerations and research challenges.

2. Big Data Technologies

A plethora of Big Data technologies have been proposed [Alexandrov et al. 2014, White 2012, Borkar et al. 2011, Zaharia et al. 2010]. In this section, we briefly describe some of those technologies and provide a comparison between them according to some categories of problems.

2.1. Overview of Big Data Technologies

This section presents an overview of the most widely used and recently discussed Big Data technologies [Sakr 2016, Singh and Reddy 2015]. To this end, we examine the main features of YARN/Hadoop, Spark, Flink and Hyracks/ASTERISK.

In many Big Data scenarios, Apache Hadoop has become the data and computational *de facto* standard for sharing and accessing data and computational resources [Vavilapalli et al. 2013]. Hadoop is a scalable open-source computation framework that allows the partitioning of computation processes across many host servers which are not necessarily high-performance computers [White 2012]. It has two main components: a MapReduce execution engine and a distributed file system (DFS) called HDFS – Hadoop Distributed FileSystem. The advantages of Hadoop mainly lie in its high flexibility, scalability, low-cost, and reliability for managing and efficiently processing a large volume of structured and unstructured datasets, as well as providing job schedules for balancing data, resource and task loads. Hadoop evolved to YARN – Yet Another Resource Negotiator, whose architecture decouples the programming model from the resource management infrastructure and delegates many scheduling functions to per-application components [Vavilapalli et al. 2013].

Apache Spark [Zaharia et al. 2016] is a unified engine for distributed data processing. It has a programming model similar to MapReduce but extends it with a data-sharing abstraction called Resilient Distributed Datasets, or RDDs. Using this extension, Spark can capture a wide range of processing workloads that previously needed separate engines, including SQL, streaming, machine learning, and graph processing. Spark [Zaharia et al. 2010] was also designed to overcome the disk I/O limitations and improve the performance of earlier systems. The main feature of Spark is its ability to perform in-memory computations. It allows the data to be cached in memory, thus eliminating the YARN's disk overhead limitation for iterative tasks.

Apache Flink [Carbone et al. 2015] is an open-source stream and batch processing framework for distributed and high-performing applications originated from [Alexandrov et al. 2014] project. It is built on the philosophy that many classes of data processing applications, including real-time analytics, continuous data pipelines, historical data processing, and iterative algorithms can be expressed and executed as pipelined fault-tolerant data flows. Flink can run as a completely independent framework, or on top of HDFS and YARN. It leverages in-memory storage for improving the performance of the runtime execution. The main novelties of Flink in comparison to previous Big Data technologies: a distributed data flow runtime that exploits pipelined streaming execution for batch and stream workloads; exactly-once state consistency through lightweight checkpointing; native iterative processing; and a sophisticated window semantics, supporting out-of-order processing.

Hyracks/ASTERIX [Borkar et al. 2011] is a partitioned-parallel software platform designed to run data-intensive computations on large shared-nothing clusters. Hyracks includes a collection of operators that can be used to assemble data processing jobs without needing to write Map and Reduce code. Moreover, it also provides a Yarn compatible layer to run existing MapReduce jobs. The Hyracks presents a scalable information management system that supports the storage, querying, and analysis of large collections of semi-structured nested data objects. Hyracks provides performance gains over MapReduce through its more flexible user model, while also being a more efficient implementation than Hadoop for MapReduce jobs for a variety of data-intensive use cases. Hyracks also achieves fault recovery performance gains over Hadoop by offering a less pessimistic approach to fault handling.

2.2. Comparison of Big Data Technologies

Companies using the Big Data technologies are usually facing challenges like: (i) dealing with the storage of heterogeneous sources such as structured, unstructured and semistructured data; (ii) the need to discover knowledge from large and heterogeneous datasets by not only applying SQL queries, but also performing complex machine learning algorithms or graph computations; (iii) continuously receiving streams of data that must be continuously processed in order of milliseconds for (near) real-time analytics. Based on these challenges, we present a comprehensive discussion on how those frameworks are able or not to provide support for *streaming and batch processing*, *generic storage* and *data analytics*.

Batch Processing. This kind of data processing is intimately related to long-time running computation over a large volume of data, all at once, over a period. It is typically performed in tasks of ETL (Extract, Transform and Load), data aggregation, training and updating machine learning models. Hadoop was broadly adopted in batch processing due to its MapReduce implementation for distributing the data processing within a computing cluster with many nodes. Hyracks also performs batch data processing. However, Spark has become the main adopted engine for large data processing by a variety of companies[1], since it brings fast in-memory data processing capability, which overcomes the Hadoop reading and writing overheads.

Streaming Processing. In stream processing, the data is processed and the results produced strictly within specific time constraints (often in the order of milliseconds and sometimes microseconds depending on the application and the user requirements). For instance, Spark Streaming receives live input data streams and divides the data into micro-batches, which are processed by the Spark engine and used to generate the final stream of results in batches. Micro-batching allows handling a stream as a sequence of small batches or chunks of data. However, it can introduce considerable overhead in the form of scheduling tasks. On the other hand, Flink can deliver all of the advantages of buffering with none of the task-scheduling overhead. Flink can also perform well on real-time or near-real-time scenarios, where insights from data should be available at nearly the same moment of data generation.

Generic Storage. HDFS can store a diverse mix of structured, unstructured and semistructured data. Hyracks can consume the data from HDFS, and it also provides the data storage AsterixDB to ingest, store, index, query, and analyze mass quantities of data using a flexible data model (ADM). Spark supports integration with a wide variety of file systems, including HDFS, MapR File System, Cassandra, Amazon S3, or the implementation of a custom solution. Flink enables the integration of heterogeneous data sets, ranging from strictly structural relational data, unstructured text data and semi-structured data. It also works with HDFS and connects to various other data storage systems. Flink and Spark do not provide a primary storage solution.

Data Analytics. YARN/Hadoop actively supports several top-level projects to create development tools and to manage its data flow and processing such as Giraph, Pig, Hive, Mahout, and HBase. Spark also supports a wide range of applications, including ETL, Machine Learning (MLib), Stream Processing (Spark Streaming), and Graph computation (GraphX). Flink' stack offers libraries with high-level APIs for different use cases: Complex Event Processing (CEP), Machine Learning (FlinkML), and Graph Analytics (Gelly). The software stack of Hyracks system is composed of various interfaces for analytics as well, like SQL (Hivesterix), XQuery (Apache VXQuery), and Graph (Pregelix). Even Hyracks can efficiently execute complex distributed data-flow operations and express full relational algebras. It also exposes low-level APIs and requires a machine learning (ML) expert to reformulate its algorithms as dataflow operators [Sparks et al. 2013].

Finally, Table 1 summarizes our discussion. It provides a *short-comparison* of Big Data technologies capabilities according to the categories of Big Data problems analyzed.

[1] `https://spark.apache.org/powered-by.html`

Technology Category	**Hadoop**	**Spark**	**Flink**	**Hyracks**
Processing Type	Batch	Mini-batch	Streaming, Batch	Batch
Generic Storage	HDFS	no primary storage	no primary storage	AsterixDB
Data Analytics	SQL, ML, Graph	ETL, ML, Graph	ML, CEP, Graph	SQL, XQuery, Graph

Table 1. Summary of Big Data technologies discussed.

3. Big Data Platforms

A Big Data platform is an ecosystem of services and technologies that needs to perform analysis on voluminous, complex and dynamic data. Thus, scaling up the hardware platform becomes imminent and choosing the right hardware/software technologies becomes a crucial decision if the user's requirements are to be satisfied in a reasonable amount of time [Singh and Reddy 2015]. A set of Big Data platforms has recently emerged, including Big Data Europe (BDE) [Jabeen et al. 2017], Hortonworks [2] and Cloudera[3]. In this section, we briefly describe these Big Data platforms and provide a comparison between them according to some categories of Big Data problems.

3.1. Overview

BDE platform [Jabeen et al. 2017] developed a computing infrastructure for handling large volumes of data in a variety of formats. It addresses the requirements of simplifying use, easing deployment, managing heterogeneity and improving scalability, and facilitates the execution and integration of Big Data frameworks and tools like Hadoop, Spark, Flink and many others. The authors have decided to use Docker as packaging and deployment methodology as well as managing the variety of underlying hardware resources efficiently alongside the varying software requirements. BDE allows performing a variety of Big Data flow tasks such as message passing (via Kafka, Flume), storage (via Hive, Cassandra), analysis (via Spark, Flink) or publishing (via GeoTriples). Moreover, the platform is open-source and completely free.

Hortonworks Data Platform (HDP) is an open-source modern data architecture that delivers immediate value by slashing storage costs as it integrates Yarn into its data center, and by optimizing Enterprise Data Warehouse costs by offloading low-value computing tasks such as ETL to Yarn. Yarn allows HDP to integrate all data processing engines across the community and commercial ecosystem to deliver consistent shared services and resources across the platform. Ambari is an intuitive Web UI and a robust REST API that makes HDP management simpler, consistent and secure. Furthermore, HDP is a complete solution offering not just data processing and management, but the enterprise capabilities to match the demands of an enterprise spanning security, governance, and operations.

Cloudera was the first company to develop and distribute Apache Hadoop-based software, and it has made data analytics on Big Data more convenient and accessible to anyone interested. It integrates Hadoop with more than a dozen other critical open source projects. Cloudera created a functionally advanced system that helps to perform

[2] https://br.hortonworks.com/

[3] https://www.cloudera.com/

end-to-end Big Data workflows. Different projects compose Cloudera ecosystem for a variety of Big Data tasks: streaming processing (via Spark), message passing (via Kafka, Flume), storage (via Accumulo, Hive, Pig, HBase), analysis (via Flink, Impala), searching (via Cloudera Search) or providing an extensible and productive web GUI for users (via HUE).

3.2. Comparison of Big Data Platforms

Following we present a set of recurrent problems usually faced by organizations that might become more complex when dealing with Big Data: (i) integrate different Big Data sources and provide a transparent view to the users; (ii) manage and protect the organization's data assets in order to guarantee generally understandable, correct, complete and secure corporate data; (iii) monitor the data, resources and applications to review and evaluate the health and performance of the whole system. Each challenge can be summarized in one of the following categories of problems: *Data Integration, Data Governance, and Monitoring Services*. In what follows, we provide a comparison between the Big Data platforms mentioned in this work, and what they provide to deal with such problems.

Data Integration. Data integration involves combining data from different sources and providing users with a unified view of them. HDP has partnered with Talend, a powerful and versatile open source solution for Big Data integration that natively supports Hadoop, including connectors for HDFS, HBase, Pig, Sqoop, and Hive without having to write any code. Talend also supports Cloudera Navigator. Another alternative for HDP is Oracle Data Integrator (ODI). A user can create a flow from sources to targets of different technologies, including relational databases, applications, XML, JSON, Hive tables, HBase, HDFS files, and so on. BDE platform goes further than HDP and Cloudera by comprising a Semantic Data Lake – a repository provided for processing and analysis the datasets in their original formats – named Ontario. Ontario builds a Semantic Layer on top of the Data Lake, which is responsible for mapping data into existing Semantic vocabularies/ontologies. A successful mapping process, termed Semantic Lifting, provides a view over the whole data. In this way, data can be extracted, queried or analyzed from the heterogeneous sources in the lake as if it was in a single format using a high-level query language. Another relevant component is Semagrow, a SPARQL query processing system that federates multiple remote endpoints.

Data Governance. Data Governance is a system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models which describe who can take what actions with what information, and when, under what circumstances, using what methods [Data Governance Institute 2018]. [Soares 2012] expands this definition by including policies regarding the optimization, privacy, and monetization of Big Data. Governing Big Data systems can be complex. Securing datasets consistently across multiple repositories can be extremely error-prone. Cloudera Navigator Data Management component is a fully integrated data management and security tool for the Hadoop that has been designed to meet compliance, data governance, and auditing needs of global enterprises. HDP uses Apache Atlas and Apache Ranger, which combine data classification with security policy enforcement. Apache Atlas was created as part of the Hadoop Data Governance initiative, and it offers the ability to view the cross-component lineage, providing a complete view of the data movement through some parsing engines such as Apache Storm, Kafka, Falcon, and Hive. Apache Ranger provides

Platform Category	**HDP**	**BDE**	**Cloudera**
Data Integration	Talend, ODI	Ontario, Semagrow	Talend
Data Governance	Atlas, Ranger	No support	Cloudera Navigator
Monitoring	Ambari	Prometheus, ELK stack	Cloudera Manager

Table 2. Summary of Big Data Platforms studied.

centralized security management for Hadoop. By integrating Atlas and Ranger, HDP allows companies to implement dynamic, runtime access policies that pro-actively prevent violations. BDE does not delve much into data governance since it does not address issues such as data privacy, sharing, and rights.

Monitoring. Monitoring is the process of proactively reviewing and evaluating what has been monitored (as data, resources or applications). Monitoring software helps to measure and track the data usually using dashboards, alerts, and reports. Cloudera Manager provides many features for monitoring the health and performance of the clusters components (hosts, service daemons) as well as the performance and resource demands of the jobs running on clusters. BDE distinguishes between resource monitoring and status monitoring. The former allows to follow up the health of a server or a component in the platform (CPU usage, memory usage, network I/O and disk utilization) while the latter offers insight in the status of a specific application. For resource monitoring, the tools Docker Stats, cAdvisor, Prometheus, InfluxDB, and Grafana can be useful at BDE platform. For status monitoring, BDE supports docker built-in logging and ELK stack. As part of HDP, Apache Ambari allows to plan, install and securely configure clusters of computers, by making it easier to provide ongoing cluster maintenance and management.

Finally, a summary of this section is presented in Table 2, that provides a *short-review* of Big Data platforms according to the categories of Big Data problems analyzed.

4. Conclusion

This paper surveys various Big Data technologies and platforms that are currently available and discusses their capabilities. A comparison between different technologies based on some important Big Data problems has been made. In addition, we also compare different Big Data platforms based on their support for data integration, data governance, and monitoring. By providing this guideline, we aim at helping organizations in the selection of technologies/platforms more appropriate to their Big Data problems. A future work consists of an empirical evaluation of these technologies/platforms by using different Big Data scenarios/applications. Moreover, we intend to compare them according to other categories of Big Data problems, such as how these platforms/technologies manage and integrate different data analysis outputs and algorithms.

Acknowledgments

This work has been supported by FUNCAP SPU 8789771/2017 research project and CAPES fellowship.

References

Alexandrov, A., Bergmann, R., Ewen, S., Freytag, J.-C., Hueske, F., Heise, A., Kao, O., Leich, M., Leser, U., Markl, V., et al. (2014). The stratosphere platform for big data analytics. *The VLDB Journal*, 23(6):939–964.

Begoli, E. and Horey, J. (2012). Design principles for effective knowledge discovery from big data. In *Joint ICSA and ECSA*, pages 215–218.

Borkar, V., Carey, M., Grover, R., Onose, N., and Vernica, R. (2011). Hyracks: A flexible and extensible foundation for data-intensive computing. In *ICDE*, pages 1151–1162.

Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., and Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society TCDE*, 36(4).

Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J. M., and Welton, C. (2009). Mad skills: new analysis practices for big data. *Proceedings of the VLDB Endowment*, 2(2):1481–1492.

Data Governance Institute (2018). Definitions of data governance. `http://www.datagovernance.com/adg_data_governance_definition/`. Accessed: 2018-05-01.

Dresner Advisory Services (2017). Big Data Analytics Market Study.

Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F. B., and Babu, S. (2011). Starfish: a self-tuning system for big data analytics. In *CIDR*, pages 261–272.

Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., and Nguifo, E. M. (2018). An experimental survey on big data frameworks. *Future Generation Computer Systems*.

Jabeen, H., Archer, P., Scerri, S., Versteden, A., Ermilov, I., Mouchakis, G., Lehmann, J., and Auer, S. (2017). Big data europe. In *EDBT/ICDT Workshops*.

Sakr, S. (2016). *Big data 2.0 processing systems: a survey*. Springer.

Singh, D. and Reddy, C. K. (2015). A survey on platforms for big data analytics. *Journal of Big Data*, 2(1):8.

Soares, S. (2012). *Big data governance: an emerging imperative*. Mc Press.

Sparks, E. R., Talwalkar, A., Smith, V., Kottalam, J., Pan, X., Gonzalez, J., Franklin, M. J., Jordan, M. I., and Kraska, T. (2013). Mli: An api for distributed machine learning. In *ICDM*.

Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S., et al. (2013). Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th Symposium SOCC*.

White, T. (2012). *Hadoop: The definitive guide*. " O'Reilly Media, Inc.".

Wu, X., Zhu, X., Wu, G.-Q., and Ding, W. (2014). Data mining with big data. *IEEE TKDE*, 26(1):97–107.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., et al. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65.