Distributed Mining of Big Data

Rui Machado[1]

[1]Faculty of Economics, University of Porto
ruimachado@outlook.pt

Abstract. This is a summary report based on the Big data presentation, from 20th of October 2016, made by Albert Bifet, which presented the big data scenario with its main characteristics and challenges and made a deep dive into what type of technologies are there available so that can we can build systems that are able to process these new types of large, fast and unstructured data and retrieve actionable knowledge from it. Is this report the main technologies presented by Professor Albert Bifet will be described and correlated.

Keywords: Big data, Hadoop architectures

1 Introduction

Albert Bifet is an experienced data scientist and an associate professor in big data at Télécom ParisTech. Besides working in with multiple companies in business analytics, data mining and machine learning projects, he co-founded in 2013, Apache SAMOA, a scalable advanced massive online analysis tool, a framework for building distributed machine leaning programs.

Aligned with his beliefs that real-time analytics are becoming increasingly important for different types of businesses, he presented at the Faculty of Economics a session on the state of the art technologies for building big data projects in a distributed computing architecture and thus help disseminating the capabilities of these systems.

Big data problems relate to the companies' capability of processing the huge amount of data produced inside their applications and systems and although a lot could be said from the processes and strategy level, technology plays a critical role in the success of this type of projects. There are currently multiple technologies to tackle big data collection and processing, which come in different types to answer different types of problem and scenarios. Albert gave an overview on these type of technologies, relating

its appearance to the problem they solve and examples of commercial tools available.(Bifet, 2016)

2 Big data

2.1 Characteristics

Big data can be defined from the problem perspective where we state it's a problem where data sets are so large or complex that traditional data processing applications can not deal with, or from the solution perspective, where we state that it's the sequence of data collection, processing and availability of huge volumes of streaming data in real-time.

Even though the definitions seem simple it has been changing over type with new additions or updates to it. While initially researchers when mentioning big data characteristics, used to focus only on high volumes of data, quickly they started to refer to it as a dataset with a high volume of data, a high variety being structured and unstructured as well as the high velocity this dataset was growing, nowadays a few more adjectives were added. (Datafloq, 2016)

As mentioned by Albert Bifet, three more characteristics are being taken in consideration to explain the complexity of a big data scenario, the decrease of veracity of data, which refers to the biases, noise and abnormality in data, the increase of variability, referring to the multiples meanings can data assume without warning and last the value of big data, which studies how and if the investment translates into business advantage. (Bifet, 2016)

2.2 Controversy of big data

Without questioning the potential value big data can bring to a data rich company, there are a few controversial topics that must be understood and taken in consideration when building a big data program to ensure the value of it exceeds the cost of its implementation.

Since the topic became trendy every data problem seems to be a big data one nowadays but not every data is big and we should always benchmark between traditional and big data approaches to ensure optimized solutions in terms of costs, time and resources. Not always bigger is better as we might be amplifying noise and thus distort the results which concludes that even a big data scenario may bring better results if reduced to a traditional data problem. Also with the grow of variables in a dataset, the number of fake variables also grow, again increasing the noise of a dataset. (Dignan, 2011)

Big data also creates an ethical issue, as in one side allows a real time customization of what user see in a website with contents oriented to its behavior in another this personalization can be seen as a reduction of the user's privacy. Its important to always keep customers and users aware of what data is being collected and analyzed to avoid this type of issues.

When it comes to technologies for distributed mining of big data, there is a hype in the market to sell Hadoop based systems, even though its not always the best tool for the job. When companies have such a big set of data sources and business processes that cannot be fit to a single infrastructure or when they lack specialized resources for this type of systems, they can struggle with high costs that will result in low value. (Bifet, 2016)

3 Technologies for distributed mining of big data

Big data created a lot of questions that software companies had to work on for years to deliver scalable technologies that could easily adopted by companies. Questions like how to split a big computational problem into small parallelized ones or how to assign and orchestrate tasks to workers distributed across a big cluster of machines and even how to merge the partial results given by the workers into a final result set or variable. (Bifet, 2016)

3.1 From MapReduce to Apache Spark.

These complicated questions quickly identified the need for an abstraction platform that could hide all answers to these questions from the programmer and thus allow a quicker adoption.

Different technological solutions to address these problems have been created over the last two decades. Some, are focused on batch based solutions, which allow processing high volumes of data where a group of transactions is collected over a period of time, others focused on stream based solutions, for processing a continual input, process and output of data, which must be processed in a near real time period and some hybrid allowing both cases. In the next chapters the most relevant solutions for this challenge will be presented. (Walker, 2013)

MapReduce.

In 2004 Google presented MapReduce as a challenge solution by providing a simple abstraction for the developer, transparently handling most of the details behind the scenes of distributed big data processing in a scalable, robust, and efficient manner.

The way this technology works for data processing follows a set of five steps:

- Iterate over a large number of records
- Map phase: extract object of interest from each
- Shuffle phase: shuffle and sort intermediate results
- Reduce Phase: aggregate intermediate results
- Generate final output

Although all these five steps and specifically the map, reduce and shuffle operations are important for the big data processing problem this solution is mostly interesting if the shuffle step is fully optimized from the communications perspective to minimize network costs in the process. MapReduce is a batch based technology. (Bifet, 2016)

Hadoop.

Motivated by the lack of a distributed storage system in MapReduce, in 2006, Apache released an open-source software framework for distributed storage, with its Hadoop Distributed File System (HDFS) and distributed processing of very large data sets on computer clusters built from commodity hardware with its MapReduce module. This platform brought to the existing MapReduce technology the capacity to abstract programmers from the file storage management democratizing even more the access to distributed computing. Because of its big data processing module is MapReduce, its nature is a batch based solution. (Bifet, 2016)

Apache Spark.

Motivated by the lack of a data processing architecture based in memory instead of the typical Hadoop disk based, Apache was born in 2009 at the University of Berkeley and became an Apache project in 2014. Because Hadoop was slow for Machine Learning, the need for a distributed data mining platform was also a driver for its creators. Spark is an open source engine and framework from distributed computing, similar to Hadoop but with some distinctive characteristics that make it faster, due to the fact that the first runs in memory while the last runs in disk and easier to develop thanks to its compatibility with Java, Python, Scala and R. Since its appearance Spark has evolved to become an ecosystem consisted by five different modules:

- Spark Engine, an Engine for large-scale data processing;
- SparkSQL, a Module for working with structured data inside Spark programs;
- Spark Streaming, a module for handling data streams and real time analytics;
- MLlib, a machine learning library for multiple data mining problem solving, including classification, regression or deep learning;
- GraphX, is an API for graphs and graph-parallel computation.

Spark allows both batch and streaming data processing approaches thanks to the fact that is fully compatible with Hadoop and thus MapReduce but also thanks to its Spark Streaming module its compatible with close to stream processing (As it approximates streaming by creating mini batch jobs). (Apache, 2016c)

3.2 Big data technology landscape

Although the latest three big data technologies are the most relevant ones, new solutions are now coming public, mostly to address and optimize the data streaming problem.

This problem relates to the fact that most of the devices people use nowadays have access to the internet, such as smartphones, fridges, watches and even washing machines, continuously streaming data, which open the opportunity to process these massive amounts of data in search for new insights regarding people's behavior and consumption pattern. This phenomenon is called Internet-of-Things and is generating the need for new big data tools more oriented for continuous data streams. In the next chapter some of these new tools and technologies will be presented. (Bifet, 2016)

Apache Flink.

Apache Flink is an open source platform for scalable batch and stream data processing that can be seen as an alternative to Apache Spark for both types of data processing problems. It also included, as Apache Spark, a Machine Learning library, a graph processing library as well as multiple APIs for creating a distributed big data processing application.

Main differences are at the programming level as Spark allows Python, R, Java or Scala development, Flink only supports Java or Scala and at the data streaming level as Spark Streaming transforms its jobs into mini batches while Apache Flink is by nature a true data stream system as every data element is immediately processed as it arrives.

If we have a data streaming problem, Flink will most probably bring better results as Spark its compatible with Hadoop and uses its HDFS module for storage management.

Apache Kafka.

Kafka is a composed platform with a publish-subscribe messaging system that allows to stream data like a messaging system, a stream processing system for real-time analytics and a storage system for storing streams in a distributed replicated cluster. Apache Kafka is typically used for building real-time data pipelines and streaming apps. (Apache, 2016b)

Kafka is a clearly a stream based solution that now only allows the big data processing capability in real time as it also allows us to integrate data coming from multiple systems, like IoT sensors in one single repository for analytics in real time.

Other competitors for data streams pipeline creation and transformation are arising in the market, such as Google Pub/Sub, which offers the same as Kafka but in the cloud, following the platform as a service paradigm. Google also launched Dataflow, a service similar to MapReduce, that when working with Pub/Sub, allows users to create and orchestrate big data processing tool in a server less approach. (Bifet, 2016)

Apache Beam.

Motivated by the need to orchestrate all the data processing jobs that data scientists and engineer create inside their applications, Apache is supporting and maintaining Beam, open source, unified model for defining and executing data-parallel processing pipelines. Beam, that got its name from combining Batch with Stream allows the creation of both types of big data processing applications.

Beam, evolved from Google data processing projects, including MapReduce and was first implemented as Google Dataflow, and in January 2016, Google and a number of partners submitted the Dataflow Programming Model as an Apache Incubator Proposal, under the name Apache Beam. (Apache, 2016a)

SAMOA.

There are multiple data mining problems such as clustering or creating classification models that feed from streams of data continuously being generated and changing over time, creating the need for a distributed platform to process and train these model.

Apache SAMOA is a distributed streaming machine learning (ML) framework that contains a programing abstraction for distributed streaming ML algorithms. This tool is also oriented to address the data stream processing problem but from the ML perspective. SAMOA is based the stream processing tool MOA, both founded and maintained by Albert Bifet. (Bifet, 2016)

4 Conclusion

Social phenomenon like the growing data social networks are producing or the advent of the Internet-of-things are increasing the data available to be mined and are constantly defying the existing tools and requesting for new ones and more optimized for continuous data stream processing.

The spectrum of distributed technologies for big data collection and processing is big at the moment, causing noise when it comes to select the best one to tackle a companies' problem. The set of tool presented in this report represent just a small fraction of amount of technologies available which reinforces the growing importance of first understanding the analytic problem we are facing deep enough so that we can design a solution agnostic to the technology. With the solution designed, looking for a technology that covers it its easier that trying to find a solution that fits technology.

This structures the main issue for big data and that is we should drive its projects by real problems the company is facing and real targets to fix them and never consider it as a hype needed to be internalized without understanding its value for the context.

5 Bibliography

Apache. (2016a). Apache Beam.

Apache. (2016b). Apache Kafka.

Apache. (2016c). Apache Spark.

Bifet, A. (2016). *Slides on distributed mining of big data presented at FEP MADSAD seminar*.

Datafloq. (2016). 3vs sufficient describe big-data. Retrieved from https://datafloq.com/read/3vs-sufficient-describe-big-data/166

Dignan, L. (2011). big-data vs traditional-databases can you reproduce youtube on oracles exadata. Retrieved from http://www.zdnet.com/article/big-data-vs-traditional-databases-can-you-reproduce-youtube-on-oracles-exadata/

Karr, D. (2016). Benefits of big data. Retrieved from https://marketingtechblog.com/benefits-of-big-data/

Walker, M. (2013). Batch vs real-time data processing. Retrieved from http://www.datasciencecentral.com/profiles/blogs/batch-vs-real-time-data-processing