Big Data Analytics: Analysis of Features and Performance of Big Data Ingestion Tools

Andreea MĂTĂCUȚĂ, Cătălina POPA
The Bucharest University of Economic Studies, Romania
andreea.matacuta@yahoo.com, popacatalina16@stud.ase.ro

The purpose of this study was to analyze the features and performance of some of the most widely used big data ingestion tools. The analysis is made for three data ingestion tools, developed by Apache: Flume, Kafka and NiFi. The study is based on the information about tool functionalities and performance. This information was collected from different sources such as articles, books and forums, provided by people who really used these tools. The goal of this study is to compare the big data ingestion tools, in order to recommend that tool which satisfies best the specific needs. Based on the selected indicators, the results of the study reveal that all tools consistently assure good results in big data ingestion, but NiFi is the best option from the point of view of functionalities and Kafka, considering the performance.

Keywords: *Big Data, Data ingestion, Real-time processing, Performance Functionality, Data Ingestion Tools*

1 Introduction

During the last years, the technology had a big impact on the applications and in the processing of data, and organizations have begun give more importance to data and invest more in their collection and management. Big Data created as well a new era and new technologies that allow analysis types of data like text and voice, which have a huge volume in the Internet and in other structures digital. The evolution of data is spectacular and it is very important to mention in this paper that in the past, the volume of data was at the level of bytes and nowadays the companies use a huge volume of data at the level of petabytes. Experts from the National Climatic Data Center in Asheville estimated that if we want to store all the data that exist in world we had need at least 1200 exabytes, but is impossible to pin down a relevant number. Maybe these sizes do not mean something for the people who do not have a direct contact to big data, but the volume of data is huge and it is very difficult to understand what these numbers mean.

V. Mayer-Schönberger and K. Cukier mentioned in [24] that “There is no good way to think about what this size of data means” to prove once that big data is in a continuous evolution and the future of it will be gloriously. The paper presents an analysis of the use of big data ingestion and present a research used to evidence the functionality and performance of most widely tools. We first introduce some concepts about data ingestion and the importance to choose it to process big data and we propose to do a short description for the tools used in analyze, offering some information about Hadoop ecosystem.

We then review existing three Apache ingestion tools: NiFi, Flume and Kafka in processing of big data and we will examine the differences between them and the strong parts of each of them. We want to offer systematic information of the main functionality of three tools developed by Apache: Flume, Kafka and NiFi used in data ingestion process and a detailed way how to combine the tools to improve the results for your requirements using for our research different ways to compare the tools based on performance, functionalities, the complexity. Our analysis shows that all three tools have something special, but there is not a one and only tool which address all of customer’s requirements and the combination of tools is the answer for that problem. We examine and recommend all the possible combinations based on the needs of customers.

The paper analyses the main characteristics of data ingestion tools. It provides key information about typical issues of data ingestion and about the reasons why we

DOI: 10.12948/issn14531305/22.2.2018.03

choose the three Apache ingestion tools instead others. Using a preliminary view, it is important to identify the common characteristics for tools. After that, the analysis results will be providing. We decided to examine Apache tools because Apache is very well known in developer's area and it is the most used web server software, running on 67% of web servers from entire world. According to the [3], "The name 'Apache' was chosen from respect for the Native American Indian tribe of Apache, well-known for their superior skills in warfare strategy and their inexhaustible endurance."

The paper has the following structure. Section 2 introduced the concept of data ingestion with big data, the necessity of it, including a short description for Hadoop ecosystem and a short paragraph where we offer information about each tool. Section 3 contends our research based on tools, analyzing the main characteristics for NiFi, Flume and Kafka, offers solid arguments why this paper use them instead another developed tool and an analyze based on the functionalities and performance for them. Section 4 presents the results of our research based on functionalities and performance for the tools and a detailed explanation for each result. Section 5, the conclusion section is the most important part because here we can observe the real importance of the information found in this paper and the scope of it and contains our final results.

2. Data ingestion and Hadoop ecosystem

2.1 Data ingestion concept

According to [9], "in typical ingestion scenarios, you have multiple data sources to process. As the number of data sources increases, the processing starts to become complicated". For a long time, data storage does not need additional tools to process the volume of data because the quantity was insignificant, but in last years when the concept of big data had appeared that begin to be a problem. As we mentioned in introduction, this paper analyses a new process to obtain and import data for their storage in a database or for immediate use called "data ingestion". According to [27], the term "ingestion" means a consumption of a substance by an organism, in our paper the consumption of a substance is represented by data and the organism can be, for example a database where the data are storage.

Data ingestion layer represents the initial step for the data coming from different sources, the step where they are categorized and prioritized, but it is important to note that is close the toughest job in the process of big data. N. S. Gill [26] mentioned that "Big Data Ingestion involves connecting to various data sources, extracting the data, and detecting the changed data". For unfamiliar readers, data ingestion can be explained like moving data (structured or unstructured) from their origin into a system where is easy to be analyzed and stored. In the next paragraphs of this paper we find important information about data ingestion from different perspectives: we prove the necessity of data ingestion, we note the challenges met in data ingestion, we offer information about parameters and key principles.

To finish the process of data ingestion it is necessary to use a tool that is capable to support the following key principles: network bandwidth, unreliable network, choosing right data format and streaming data

2.1.1 The necessity of data ingestion

In many situations, when using big data, the source of data structure is not known and if the companies, for example use the common data ingestion methods it is difficult to manipulate the data. For the companies data ingestion represents an important strategy, helping them to retain customers and obtain increase profitability.

The main advantages that demonstrate the necessity of data ingestion are the following:

- Increased productivity. It is taking a lot of time for companies to analyze and to move data from different sources, but with data ingestion the process is easier and the time can be used to do something else
- Ingestion of data in batches or in real time. In batches, data are stored based on periodic intervals of time

- Data are automatically organized and structured, even if there are different big data formats or protocols.

2.1.2 Data ingestion challenges

Variance and volume of data sources are in a continuously expansion. Extracting data from these sources can be extremely challenging for users, considering the required time and resources. The main issues in data ingestion are the following:

- Different formats in data sources
- Applications and data sources are evolving rapidly
- Data capture and detection are time consuming
- Validation of ingested data
- Data compression and transformation before ingestion

2.1.3 Data ingestion parameters

The main ingestion parameters used in the comparison are the following:

- Data velocity- this parameter is based on the speed to process data from different sources like human interaction, social media, networks.
- Data size- Because data ingestion works with huge volume of data, they are generated from multiple sources to increase the time
- Data Frequency-This parameter can have two ways to process data: in real time or batch
- Data Format-Every company choose different format for their data and the data ingestion needs to adapt for every situation

2.2 Hadoop ecosystem

Apache Hadoop ecosystem is an essential supporting structure for processing and storing large amount of data (Big Data). The Apache Hadoop ecosystem grows continuously and it consists of multiple projects and tools with valuable features and benefits that provide capacity of loading, transferring, streaming, indexing, messaging, querying and many others. Hadoop contains two main elements: Hadoop Distributed Filesystem (HDFS) and MapReduce. The HDFS is a file system designed for data storage and processing of data. HDFS is made for storing and providing streaming, parallel access to large amount of data (up to 100s of TB). HDFS storage is distributed over a cluster of nodes. MapReduce is a large dataset processing model. As the name suggests, it is composed of two steps. The initial step, Map, establishes a process for each single key of the records to be processed (key value type). The final step, Reduce, performs the operation of summing the results, processing the data output from the map phase according to the required operator or function, resulting in a set of value key pairs for each single key.

Swizec Teller notes in [22] that these two projects can be configured in combination with other projects into a Hadoop cluster. A cluster can have hundreds or thousands of nodes and they can be difficult to manually configure. Hadoop cluster covers the need for tools to easily and effectively configure systems and data. HDFS and MapReduce might be executed on separate servers. They are named Hadoop clients and security is the main reason for physically separating Hadoop nodes from Hadoop clients. If we are deciding to install clients on the same servers as Hadoop, we will have to provide a high level of security to every user for access. Logically and physically separating them simplifies the needed configuration steps. There are many sub-projects (managed mostly by Apache, which are made by free organizations) designed for maintenance and monitoring that very well integrate with Hadoop and they lets us concentrate for developing data ingestion rather than monitoring it. Three of the commonly used tools for data ingestion in Hadoop are rigorously described in the following sections.

2.3 Kafka

Apache Kafka is a distributed, high-throughput messaging system, a publish - subscribe environment that provide highly availability. With one broker handling hundreds of MB per second of reads and writes from several clients, Kafka is a very

DOI: 10.12948/issn14531305/22.2.2018.03

fast system. Replication is used for messages across the cluster and then stored on disk. According to [7], "Kafka can be used for stream processing, web site activity tracking, metrics collection and monitoring, and log aggregation". "It is a paradox that a key feature of Kafka is its small number of features. It has far fewer features and is much less configurable than Flume", noted Ellen Friedman and Ted Dunning in [8]. Flume will be described in the next section of this paper. They also discovered that "Kafka is similar to Flume in that it streams messages, but Kafka is designed for a different purpose.

While Flume is designed to stream messages to a sink such as HDFS or HBase, Kafka is designed for messages to be consumed by several applications". The principal elements of Kafka architecture are *Producer*, *Broker*, Consumer and *Topic*. Topics are used for feeding of messages. Producers send the messages to topics and consumers, who can subscribe to those topics and consume the messages from that topics. Topics are partitioned and is attached a key to each message and a partition is like a log.

2.4 Flume

"Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data." is a reliable definition found on official website for Apache Flume. This paper analyses the newest version for tools and the last version stable for Flume is 1.8.0, the eleventh release for apache project that offer to the users a stable product, compatible with older versions of the Flume (1.x code line) and it is a software ready for production. This tool is made to ingest and collect huge volumes of data from multiple sources into Hadoop Distributed File System (HDFS), the most used types of data for processing are sensor and machine data, social media data, maps, astronomy, aerospace, application logs or geo-location data. We observe the using of Flume in one specific example where the tool is used for the logging of manufacturing operations, the log is generated in every run of the product when it comes off the line and generate a file with information about the respective run. In a day the product runs for thousands of times and generate a large volume of data stored in log files and using Flume, data can be stream into a tool for analyze, like we can see in the image below, followed by storage process of them in HDFS. We remark that in general, Flume allows users to ingest and store into Hadoop for future analyses, data from multiple sources and with different sizes, use horizontally scale to ingest data, the user has the guarantee that his data are delivered based on the transactions between agents, use the insulate system in the situation when incoming data rate is bigger than a standard rate and it has a better integrated bond with Hadoop ecosystem in contrast to Kafka or NiFi.

J. Kim and B. Bengfort note in [11] that the data flows in Flume like a pathway which ingest data from origin to destination. Data or events are moved from source to destination based one sequence of hops and the concept is named Flume agent (a JVM process) which consist three important components: channel, sink and the source as we can see in the below image.

Source represents the part of the Agent where data are received from data generators, followed by transfer them to the channels from Flume events, Channel can work with different sources and sinks and are represented a bridge between the sinks and the sources: receives the data from the source and use the buffer till they are consumed by sinks. Sink represents the final component of the Flume agent where the data from the channels are consumed and send to the destination (the data are stored in HDFS at this step). We note that the biggest disadvantage of using Flume is that the data can be lose in a very easy way, for example if the user choose the Memory channel with high throughput, when the agent node goes down the data will be lost.

2.5 NiFi

Some systems are generating the data and other systems are consuming it. Apache NiFi is developed for the automation of this flow. In [16] Apache NiFi is defined as "a data flow

DOI: 10.12948/issn14531305/22.2.2018.03

management system that comes with a web User Interface that helps to build data flows in real time, it supports flow-based programming and the graph programming includes processors and connectors, instead of nodes and edges". The user connects processors together with connectors and the data will be defined how to be manipulated. A strong feature is Nifi's capability of ingesting any data using ingestion methodologies for any particular data. Similar to inputting data. The output is very customizable too. T. John and P. Misra observed in [23] that "The Apache NiFi website states Apache NiFi as - An easy to use, powerful, and reliable system to process and distribute data". We consider that is a good alternative of Apache Flume having a vast set of features and easy to use web user interface. It is easy to set-up and it is very highly customizable.

3. The research methods

This section contains information about the analysis of the main characteristic for tools: Flume, NiFi and Kafka and represents our analysis of the functionalities and characteristics with the scope to put in evidence the choice of them for creating the content of this paper. Our analysis is based on the comparison of them from different perspectives like performance, functionality, necessity. First, when we searched information about data ingestion tools we found over 30 different tools used by companies in this process and we were in the situation to choose the best of them for our analysis. The choice of the data ingestion tool for a company depends on multiple factors such as target, transformations (simple or complex), data source, performance, necessity so we used the same criteria in our research.

Next, we searched for articles and opinions that contained the key words like "data ingestion used tool", "first option of data ingestion tool" and we obtained the main used tools for data ingestion. The final decision was based on [18] were, based on top 18 data ingestion tools, Flume is on second position, followed by Apache Kafka and Apache NiFi, first option been Amazon Kinesis. Based on this top we decided that our paper will analyze three tools which represent a main choice for companies and users. We preferred to use in our research Apache tools mentioned in this paper because an advantage of them is the possibility to combine them for a better result. Comparing with the other tools from data ingestion area, we noted that the analyzed tools from this paper have some special characteristics such as the guarantee that they are reliable offering zero data loss, using large volumes of data Apache Kafka and Flume systems provide scalable, reliable and high-performance. In our decision we based on criteria which convinced us that if we are put in a hypothetical situation to choose a tool for data ingestion we will use one of them.

The last criteria and maybe the most important which helped us to decide on this choice was the information from articles and books such as [16], [17] and [21], based on using of this tools in known application. We find that the main criteria when a company wants to choose a tool for data ingestion are: speed to ingest data in a rapid way, platform support which offers the facility to connect with data stores, the facility to scale the framework to work with large datasets and the facility to extract and access data from sources without impact on their ability to execute transactions or performance and in our choice for NiFi, Kafka and Flume we used that criteria.

3.1 Functionalities of analyzed tools

An important part of the study is the analysis of tools functionalities. In this subsection we present the functionalities used for tool comparison. We analyzed the following indicators: reliability, system requirements, limits of the tool, stream ingest and processing, guaranteed delivery and data type. In the following paragraph we will justify our choice for these indicators. In the next paragraph, we will present the results. In the result section we examine the functionality for each tool using a standard scale, with three values: complete implemented functionality, partial implemented functionality and no implemented functionality.

According to [25], “data collection and transportation should be reliable with minimum data loss”. In our research, reliability was based on the ability to deliver events and logging them in situation when one of the tools presented have a software crash, scarce memory or bandwidth and the possibility to lose data can appear. We consider that guaranteed delivery functionality is very important because user needs to have the guarantee that his data are completed after data ingestion process. Limits of the tools were included in our analysis to expose their disadvantages.

We want to note in this paper the fact that we consider that from this point of view a perfect tool does not exist and for companies can be a better solution to combine them. Another important functionality used in our investigation was data type because it is important for the user to know what type of data can ingest with the chosen tool.

3.2 Performance measurements of the selected tools

In performance testing using big data are included two main actions: data ingestion and throughout where is verified how the fast system can consume data from various data source and data processing which involves to verify the speed with which the map or queries can reduce jobs in execution. We note in our research performance measurements because we consider that the user needs to know information about speed, the number of processed files per minute, the acceptable size for the files for the tool. In our analyze we based on performance measurements studies created on the tools official pages and on the opinions found in articles, from different users who use Kafka, NiFi or Flume. In this paper, we used for our research in performance the following indicators: speed, number of processed files per second, scalability and message durability.

For businesses can be challenging to ingest big data at a reasonable speed or to process it efficiently with the scope to maintain a competitive advantage so the speed indicator needs to be included in our analysis. Another indicator included in our research was the number of processed files per second. We want to note the fact that for our tools the number of processed data per second differed because of the size of files. Scalability was included in our research based on what Cory Isaacson said in [5]: “When managing a successful expanding application, the ability to scale becomes a critical need. Whether you are introducing the latest new game, a highly popular mobile application, or an online analytics service, it is important to be able to accommodate rapid growth in traffic and data volume to keep your users happy.”. We consider that indicator an important one because data ingestion tools work with a huge volume of big data and the scalability can help in this process. Message durability was included in our research to make sure that even if the tool dies, the task is not lost. When one of tools quit or crash the information about queues and messages are forgot and to make sure that messages are not lost it is necessary to mark both the and messages queue as durable. In the result section we examine the performance indicators for each tool using a standard scale which can have three values: high, medium and slow.

Bellow, we present a performance measurements study for Flume, presented on the Flume’s official page and we used it in our research because it is the most explicit study found on this subject which evidence performance for Flume in big data ingestion process. Test was made by M. Percy in [15] who used the following test setup: Flume agent was run in a single JVM on his own physical machine and a separate client machine was used to generate load in syslog format against the Flume box. Data was store by Flume onto a 9 node HDFS cluster which was configured on a separate hardware. In this test, virtual machines were not used. On the point of view of hardware specifications CPU used was Intel Xeon L5630 2 x quad-core with Hyper-Threading @ 2133MHz (8 physical cores), memory: 48GB and operation system SuSE Linux 64-bit. For Flume configurations, Mike Percy used 1.6.0u26 java version, one agent, for channel used Memory Channel and

DOI: 10.12948/issn14531305/22.2.2018.03

for sink HDFSEventSink with Avro event serialization and snappy serialized compression. For data description he used event size 300 bytes. According to the obtained results, Flume is capable to assure an approximate average of performance equal with 70000 events/second, on a single one machine, without data loss during the test.

4. The research results

4.1 The functionality comparison results

Based on the results obtained in our research, we compare the tools from the functionality point of view for a better analyze (see Table 1). According to definition from Wikipedia for Apache Flume, "is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data" in our research we can prove that this functionality is completed implemented and Flume have a fail over mechanism that can move the data flows on a new agent without exterior interventions. We consider that the best choice for this functionality is Kafka because in situation when a single point failure data is available in contrast of Flume where the user cannot access events till the disk is recovered. On the other hand, Nifi is reliable throw definition, but in real world is better to combine this tool with Kafka for using Kafka's reliable data stream storage.

Flume and Nifi represent the main choices for data guarantee delivery in comparison with Kafka where that is not guarantee in totality. Depending on protocol, NiFI allows supports guaranteed delivery with the mention that supports most once or at least once and Flume guarantee the delivery of the Events using a transactional approach. All of them have a limitation for functionality "limits of the tools" because a one and only tool does not exist to be addressed for all requirements or to do everything. In our analyze we obtained a list of limits for every tool. For Flume we obtained that when Kafka Channel is used the possibility to loss data appear, data are limited at kb dimension and the data replication does not exist. On the other hand, Kafka has the same problem as Flume for dimension of data (kb), it has fixed protocol, format and schema, a custom code is often needed. The last one, NiFi does not accept data replication and it is not a variant to use for CEP or windowed computations. Data type for Flume is represented by the next file formats Sequence File, DataStream or Compressed Stream, Kafka accept data type like JSon, PoJo or Java bean and the fastest way: arrays, Nifi uses data object (Flow File). Conclusion for this functionality is that a tool that can process all types of data does not exist and the decision for the user depends on his needs. We consider that Nifi is the best option from the point of view of system requirements because can run on laptop and can be used with a cluster across enterprise class servers, hardware and memory needed depends on the size of data, can run on Windows, Linux, Unix or Mac OS, support all type of browser and requires java 8 or newer. On the other hand, Flume can run only on Linux, requires java 8 or newer, requires sufficient space on disk for sinks and channels and the agents need permission to Write/Read directories. Kafka requires machines with a lot of memory on them, can run on Unix or Windows, requires java 8 or newer. Using Hadoop, Flume can be used to transfer, collect, aggregate streaming events because it is a distributed system, while simple, flexible and intuitive programming model is based on streaming data flows. Our analysis provides the fact that Flume maintains a main list of ongoing data flows. In Kafka, messages are put into topics, which are split into partitions and this one is replicated across the nodes in the cluster. Kafka provide a huge throughput persistent messaging which is used to scalable and allow parallel data loads for Hadoop. NiFi provides real-time control and it is easier to manage when run in a cluster the movement of data between source and destination. In conclusion from the point of view of functionality, according to our analyze we consider that NiFi represents the best solution to use in a company.

Table 1. Functionality tools comparison

Functionality	**Flume**	**NiFi**	**Kafka**	**Recommended tool**
Reliability	Partial implemented functionality	Partial implemented functionality	Complete implemented functionality	Kafka
Guaranteed delivery	Complete implemented functionality	Complete implemented functionality	Partial implemented functionality	Flume and NiFi
Data type	Partial implemented functionality	Partial implemented functionality	Partial implemented functionality	Flume, NiFi and Kafka
System Requirements	Partial implemented functionality	Complete implemented functionality	Partial implemented functionality	NiFi
Stream ingest and processing	Partial implemented functionality	Complete implemented functionality	Partial implemented functionality	Flume, NiFi and Kafka
Limits of the tool	Partial implemented functionality	Partial implemented functionality	Partial implemented functionality	Flume, NiFi and Kafka

4.2 The performance comparison results

In the table 2, we did a summary for performance indicators for Kafka, Flume and NiFi based on the results of our analysis. To determine the measure for tools in our research we based on the functionality results where we note that Flume and Kafka have a limit for data size (KB), so from this point of view Flume and Kafka are the best options for the number of files processed per second because the size of them is small. Speed of tools was put to medium for all of them because the indicator does not have a standard limit, it depends on the needs of the user and if the user wants to increase it can use tuning procedure. According to [14], "Kafka is a general purpose publish-subscribe model messaging system, which offers strong durability, scalability and fault-tolerance support." we consider that this tool is the best option for scalability in comparison with NiFi and Flume which are distributed, reliable, and available systems. Kafka is very scalable and one of the key benefits of it is that adding a large number of consumers can be made in an easy way without down time or affecting performance in comparison with Flume or NiFi where this process cannot be made in an easy way. For the last one indicator we obtained that Flume represents one of the best choice because it supports multiple interceptors chaining and data flow models and with them, flume makes event transforming and filtering very easy and Kafka supports replication synchronous and asynchronous based on the durability requirement and uses commodity hard drive. On the other hand, NiFi do not have support native for message processing and in this case the tools need to integrate with other event processing frameworks to complete the job. In conclusion our choice from the point of view of performance indicators is Kafka because it obtained good results for processing, message durability and scalability.

DOI: 10.12948/issn14531305/22.2.2018.03

Table 2. Performance indicators

Performance indicator	**Flume**	**NiFi**	**Kafka**	**The best choice**
Speed	Medium	Medium	Medium	All of them
Number of files processed per second	High	Medium	High	Flume and Kafka
Scalability	Medium	Medium	High	Kafka
Message durability	High	Low	High	Flume and Kafka

5. Conclusions

The complexity and volume of data generated by human and machines activity is increasing continuously. This paper presented an analysis of the use of big data ingestion and the process of ingesting the variety, volume and veracity of Big Data.

After introducing the concept of data ingestion with Big Data, the necessity of it, and realizing a short description for Hadoop ecosystem and a description of three of most widely tools for big data ingestion, we examined these three Apache tools NiFi, Flume and Kafka in order to determine the common characteristics and analyze the parts of each other according performance and functionality. This research showed that in terms of performance, Kafka offers the best results, and in terms of functionality, Nifi is the best option.

References

[1] A. Holmes, *Hadoop in Practice, Second Edition*, Manning, 2014

[2] Ali-ud-din Khan M., M. Fahim Uddin, N. Gupta, "Seven V's of Big Data," in Proc. of Zone 1 Conference of the American Society for Engineering Education, 2014

[3] Apache Hadoop. (2018). What Is Apache Hadoop? Retrieved from http://hadoop.apache.org/

[4] B. Bengfort, J. Kim, *Data Analytics with Hadoop: An Introduction for Data Scientists*, O'Reilly Media, 2016

[5] C. Isaacson, *Understanding Big Data Scalability: Big Data Scalability Series, Part I*, p. 30, Prentice Hall, 2014

[6] D. Zburivsky, *Hadoop Cluster Deployment*, 2013

[7] D. Vohra, *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools*, Apress, 2016

[8] E. Friedman , T. Dunning, *Real-World Hadoop*, O'Reilly Media, 2015

[9] H. Shah, N. Sawant. *Big Data Application Architecture Q&A: A Problem - Solution Approach*, Apress 2013

[10] J. Kreps, Putting Apache Kafka to Use. A Practical Guide to Building a Streaming Platform, Retrieved from: https://www.confluent.io/blog/stream-data-platform-1/, 2015

[11] J. Kim, B. Bengfort, *Data Analytics with Hadoop*, O'Reilly Media, Inc., 2016

[12] J. Shan, Why Big Data is a big deal, 2014

[13] K. Noyes, "How Apache Kafka is greasing the wheels for big data," Computerworld, Nov. 1, 2015

[14] L. Jiang, Flume or Kafka for Real-Time Event Processing. Retrieved from https://www.linkedin.com/pulse/flume-kafka-real-time-event-processing-lan-jiang, 2015

[15] M. Percy, Flume NG Performance Measurements. Retrieved from https://cwiki.apache.org/confluence/display/FLUME/Flume+NG+Performance+Measurements, 2012

[16] N. Kumar, P. Shindgikar, *Modern Big Data Processing with Hadoop*, Pack Publishing, 2018

[17] N. Garg, *Learning Apache Kafka - Second Edition*, Packt Publishing, 2015

DOI: 10.12948/issn14531305/22.2.2018.03

[18] Predictive Analytics Today. (2017). Top 18 data ingestion tools. Retrieved from https://www.predictiveanalyticstoday.com/data-ingestion-tools/

[19] R. Manivannan. *Using Kylo for Self-Service Data Ingestion, Cleansing,and Validation*, 2017

[20] S. Ornes, Explainer:Understanding size data. Retrieved from https://www.sciencenewsforstudents.org/article/explainer-understanding-size-data, 2013

[21] S. Hoffman, *Apache Flume: Distributed Log Collection for Hadoop Second Edition*, p. 56-70, Packt Publishing, 2015

[22] S. Teller, *Hadoop Essentials*, Packt Publishing, 2015

[23] T. John , P. Misra, *Data Lake for Enterprises*, Packt Publishing, 2017

[24] V. Mayer-Schönberger, K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*, Eamon Dolan/Mariner Books, 272 p., ISBN: 978-0544227750, 2014

[25] Y. Yang, Data ingestion overview. Retrieved from https://medium.com/@Pinterest_Engineering/scalable-and-reliable-data-ingestion-at-pinterest-b921c2ee8754, 2017

[26] N. S. Gill, Data Ingestion, Processing and Architecture layers for Big Data and IoT, https://www.xenonstack.com/blog/big-data-engineering/ingestion-processing-big-data-iot-stream/, 2017

[27] Ingestion – Wikipedia, https://en.wikipedia.org/wiki/Ingestion

Andreea MĂTĂCUȚĂ has graduated the Faculty of Cybernetics, Statistics and Economic Informatics from the Bucharest University of Economic Studies in 2016. She graduated the master at the same faculty, in Economic Informatics and this year (2018) she got her master diploma. Currently she has a full job as programmer using Java at the Axway Company and in the past she worked on C#. Her main focus is to learn more about programming and to become better on his domain. About her personal life she is passionate about technology, reading, pets, sport.

Catalina POPA has graduated the Faculty of Electronics, Telecommunications and Information Technology from Polytechnic University of Bucharest in 2016. She graduated in June 2018 the master of Economic Informatics at Faculty of Economic Cybernetics, Statistics and Informatics from Academy of Economic Studies. Currently she is a Business Intelligence Application Administrator at Orange, in IT Services Operations Department. Her work focuses on improving the monitoring systems by adding and modifying Shell and PL/SQL scripts

DOI: 10.12948/issn14531305/22.2.2018.03