A Systematic Mapping Study for Big Data Stream Processing Frameworks

Mohammed Alayyoub
Atilim University
Turkey

Ali Yazici
Atilim University
Turkey

Ziya Karakaya
Atilim University
Turkey

***Abstract*—The choice of the most effective stream processing framework (SPF) for Big Data has been an important issue among the researchers and practioners. Each of the SPFs has different cutting edge technologies in their steps of processing the data in motion that gives them a better advantage over the others. Even though, these technologies used in each SPF might better them, it is still difficult to conclude which framework berforms better under different scenarios and conditions. In this paper, we aim to show trends and differences about several SPFs for Big Data by using the so called Systematic Mapping (SM) approach using the related research outcomes. To achieve this objective, nine research questions (RQs) were raised, in which 91 studies that were conducted between 2010 and 2015 were evaluated. We present the trends by classifying the research on SPFs with respect to the proposed RQs which can direct researchers in getting an state-of-art overview of the field.**

***Index Terms*—Big Data, Streaming Frameworks, Storm, Flink, Spark, S4, InfoSphere**

I. Introduction

RECENTLY, the Big Data technology has been gaining increasing popularity. The term ”Big Data” is generally used for the capability of storing, managing and processing vast amounts of disparate data sets that cannot be handled using classical techniques. For Big Data, Apache Hadoop provides an open-source, reliable and distributed system. Hadoop includes (a) Hadoop Distributed File System (HDFS) for storage; (b) YARN: A framework for job scheduling and cluster resource management; (c) MapReduce: A YARN-based system for parallel processing of large data sets. [1]; (d) Hadoop commons is the other utilities provided within the system. These components have proven themselves for batch oriented processing in Big Data. Although Hadoop can handle such jobs effectively, the necessity for non-batch oriented jobs like real-time processing, and iterative jobs has been on the rise.

The Big Data community has leaned towards creating other solutions, such as developing frameworks for real-time processing, for example Apache Flink [2] which is used to be called Stratosphere, Apache Spark [3] and Apache Storm [4]. Each of these frameworks can work in a Hadoop environment and has a different way of processing data for both batch and stream oriented jobs. Also, Apache S4 [5] was developed by Yahoo, and then contributed to Apache Software Foundation[6].

Real-time stream processing is really more difficult than batch processing, because there are additional considerations one needs to consider such as, latency, in-memory computing since real-time processing cannot tolerate storing and analyzing data on disks all the time in an effective manner. Also, real-time stream processing needs to handle vast amount of disparate stream data from a variety of sources in time.

Data streams are now very common; they include log, click, message, sensor, and event streams and even streaming data that come from social media usage. All such data allow people to take real-time action, including (1) taking action in the moment of equipment malfunctioning through analyzing sensor and log streams, (2) providing real-time recommendations to the users by taking account of other users that show a relevant pattern, (3) detecting a fraudulent transaction, say, in a user's bank account, by analyzing the user's pattern of transaction history.

All these capabilities are not easily provided and each of the streaming frameworks in big data has a different way of handling issues of scalability, latency, fail recovery and processing vast amount of disparate stream data. Additionally, in the Big Data community, batch-oriented frameworks have matured adequately in a place where streaming frameworks are not so yet. The streaming area in Big Data is still a very hot topic now and it has a long way to be mature.

This article proposes an SM study which has differences from literature reviews. In an SM approach, the study maps out and categorizes existing literatures in a research field in order to characterize quantities in several aspects. At the end, the study identifies gaps and trends, and illustrates them with graphical views. However, literature reviews examine recent or current literatures of a field and typically provides narrative results [7].

In this SM study, the authors had to limit the study to a handful of Big Data streaming frameworks. Otherwise the paper pool would include various types of frameworks having different purposes which would make it harder to find common grounds for constructing the classification scheme and, as such mitigate the validity of the present study. As a result, we try to find the gaps and the trends by classifying the research on the most popular stream processing frameworks, namely, Apache Flink, Apache Spark, Apache Storm, Apache S4 , and IBM's InfoSphere [8].

The remainder of this paper is structured as follows. A review of the related work is presented in Section 2. Section 3 explains the research methodology for this SM study. Section

M. Alayyoub was M.S. student at Institute of Natural and Applied Sciences, Atilim University, Ankara-Turkey, e-Mail: muhammedalayyoubb@gmail.com

A. Yazici was with the Department of Software Engineering of Atilim University, Ankara, Turkey. Email : ali.yazici@atilim.edu.tr

Z. Karakaya was with the Department of Computer Engineering of Atilim University, Ankara, Turkey. Email : ziya.karakaya@atilim.edu.tr

Manuscript received April 19, 2005; revised August 26, 2015.

4 defines the outcomes of applying the research methodology. Section 5 presents the results of our SM study. Finally, Section 7 summarizes the main findings and trends.

II. Related Work

Here a brief overview of existing secondary studies focusing on SPFs of Big Data is provided.

A survey is conducted on the frameworks for distributed computing including Hadoop, Spark and Storm through summarizing their architecture and work-flows [9]. The paper defines Spark as the newest player in the MapReduce field by making data analytics fast to write and to run through in-memory computation. In the same paper, Storm is mentioned as being the Hadoop of Real-time Processing and specified as a complement to Hadoop rather than an actual replacement. The paper also presents a brief summary of comparison between Spark streaming and Storm.

In [10], a survey on modern approaches for Big Data stream processing is presented. It provides a brief overview of what stream processing is, while also mentioning about several challenges faced with it and then discussing several solutions by briefly examining about the existing frameworks in the area which including Apache Storm, Spark Streaming, Apache Samza, Apache Flume, Amazon Kinesis and IBM InfoSphere Streams.

Another study presents a survey [11] of the open source technologies that support Big Data processing in a real-time fashion, including their system architecture and platforms. It discusses how to leverage lambda architecture on stream processing systems and introduces several stream processing systems including, Hadoop Online, Spark and Spark Streaming, Storm, Flume, Kafka, Scribe, S4, HStreaming, All-Rite, Impala. Additionally, the paper presents a comparative summary of each of the systems.

A literature review is represented about approaches towards big data parallel processing and distributed computation in [12]. The article also presents information about the real-time stream computing system, S4, how discretized streams are used in Spark streaming.

In [13], 8 real-time stream processing requirements are outlined to provide high-level guidance for evaluating alternative stream processing solutions.

In [14], a literature survey and system tutorial for big data analytic platforms is presented. For our purposes, the article presents a comparison of streaming and batch processing in different aspects. Additionally, the article presents another comparison about S4 and Storm.

All these studies present brief overview or comparison of several stream processing frameworks within Big Data environment. However, our SM study intends to portray the interest of the research community about the frameworks by depicting trends and gaps in several aspects.

III. Research Methodology

This SM research follows the guidelines proposed by [15]. Basically, SM is a defined method to build a classification scheme and structure for a field of interest. The analysis of results focuses on frequencies of publications for categories within the scheme. The essential process steps of our SM study involves: (i) defining the RQs, (ii) conducting the search for relevant papers, (iii) applying the inclusion and exclusion criteria on the papers in the pool, (iv) developing a classification scheme depending on the RQs. In the remainder of this section, these steps are explained.

A. Goal and Research Questions

Our SM study aims to show trends and differences by classifying the research on SPFs. To achieve our goal, we put forward the following 9 RQs as follows:

RQ 1. What types of contributions are made by the papers?

RQ 2. What type of research methods are used in the papers?

RQ 3. What type of research methods are used for each of the framework in the papers?

RQ 4. What is the annual number of publications for each Big Data stream processing framework?

RQ 5. What is the ratio of experimentation type (batch only, stream only or both) used for each Big Data stream processing framework in the papers?

RQ 6. What is the ratio of contribution purposes (usage enhancement, performance enhancement or both) for each Big Data stream processing framework in the papers?

RQ 7. Which data ingestion source is used most for each framework?

RQ 8. What is the most preferred range for the number of nodes used in experimentation for each Big Data stream processing framework?

RQ 9. What type(s) of data is used most for each Big Data stream processing framework?

B. Search Strategy

The authors searched the following five academic paper search engines to find relevant papers: (1) IEEE Xplore, (2) ACM Digital Library, (3) CiteSeerX, (4) Science Direct, and (5) Google Scholar.

In order for authors to ensure that not leaving out any of the relevant papers about the Big Data streaming frameworks in question, several search strings were used as shown in Table 1.

C. Inclusion and Exclusion Criteria

After applying the exclusion criteria listed in Table II any paper containing at least one of the Big Data frameworks is included covering Apache Spark, Storm, Flink, S4 and IBM's InfoSphere Streams. Also, to ensure the inclusion of all relevant publications to the pool, the authors researched

TABLE I
SEARCH STRINGS USED TO FIND RELEVANT PAPERS

Search Strings	
1. [(Apache) AND (Spark OR Storm OR Flink OR S4)]	4. [”IBM’s InfoSphere” OR ”InfoSphere Streams” OR InfoSphere]
2. [Spark OR Storm OR S4 OR ”Yahoo’s S4” OR Flink OR Stratosphere]	5. [”Big Data” AND (stream OR real-time)]
3. [(Stream OR Real-time) AND Processing]	6. [”Big Data Streaming” AND (framework OR platform)]

for: (1) Related papers referenced from those already in the pool, and (2) Related papers from major Big Data (e.g. Big Data, SIGMOD) research venues. Additionally, we used the exclusion criteria in the table below to screen irrelevant papers.

D. Classification Scheme

While formulating the RQs, the authors tried to find common grounds in all papers where each might show differences in choice, so that each paper can answer RQs by signifying a trend. This approach helped us to create the classification scheme shown in Table III.

The term contribution facet for the classification scheme is taken from [15]. The term describes the types of contributions such as being a method/technique/approach, tool or model. The author also added another contribution types such as: Framework, Architecture, platform, Empirical (Case) Study, Analyze, Comparison, Overview, Others.In the value set, Analysis means the article analyzes one of the stream processing framework by evaluating it with experiments (benchmarks). Also, Comparison articles compare one of the selected Big Data stream processing framework with each other or any other software.

The term ”research facet” denotes the type of research approach used in each paper where the guidelines for this category are taken from [15]. The research facets adopted for this study are listed and summarized [16] below:

Solution Proposal: A solution for a problem is proposed, and the solutions can be shown by a good line of argumentation or a small example.

Validation Research: Techniques investigated have not yet been implemented in practice and are novel. Possible research methods are experiments, simulation, and prototyping.

Evaluation Research: Techniques are implemented in practice and an evaluation of the technique is conducted. The contributions are studied empirically, such as case study, or field experiment. Also, the conclusions need to be supported in the papers.

Experience Papers: Experience papers have to be the personal experience of the author and they explain on what and how something has been done in practice.

IV. EXECUTION OF RESEARCH METHODOLOGY

This section presents the outcomes from applying the steps of the research methodology. After defining the RQs, the search strings are used to find relevant papers. The search strings are modified according to different syntax of the academic search engines. The search strategy provided us 451 candidate studies from the selected sources. Some of the papers are eliminated through using exclusion criteria. Also, the introduction and conclusion sections of the papers are taken into consideration in case of uncertainities. This whole elimination step dropped down the number of the relevant articles to 91. Then, the data is extracted from the papers using the defined classification scheme.

V. RESULTS OF SYSTEMATIC MAPPING

The results of our SM study are illustrated for each of the RQs.

RQ 1. What types of contributions are made by the papers?

Figure 1 shows the distribution of the type of papers by contribution facets for the 91 papers included in this study. Some of the papers were classified under more than one facet based on their contributions. For example [17] proposes two contributions: (1) a framework and (2) a comparison.

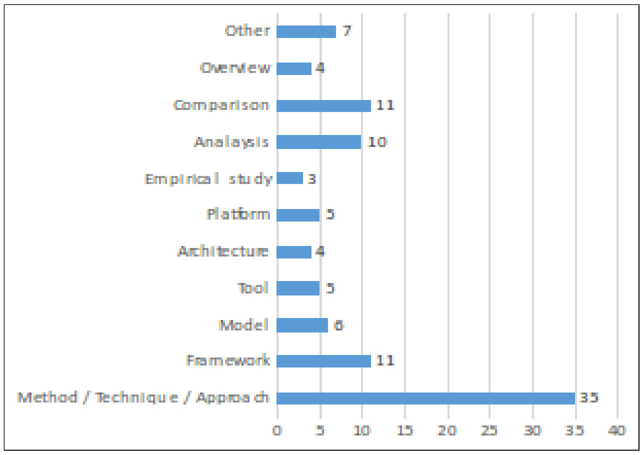

Fig. 1. Contribution facets of the papers

Figure 1 indicates that proposing new methods, techniques or approaches has attracted the most of researchers with 35 times. Also, proposing new frameworks, as well as comparing and analyzing the existing ones have taken high portion with the count of papers respectively 11, 11, 10. There were 7 papers which could not be categorized into the contribution facet categories, thus the author categorized them under ”Other”.

RQ 2. What type of research methods are used in the papers?

Figure 2 depicts the distribution of the type of papers by research facet. Approximately half of the papers are validation research (39 out of 91). Also, the papers with evaluation

TABLE II
EXCLUSION CRITERIA

#	Criteria
1	Papers that do not include any content about Spark, Storm, Flink, IBM's InfoSphere, S4.
2	Papers written in a language other than English.
3	Short papers which do not contribute - provide benefit - to our SM study.
4	Work in progress.
5	Papers that tutor one of the tools in a way that does not contribute to our SM study.
7	Duplicated studies such as those published in other papers. In such as case; we included the most recent one.
8	The papers which are not freely accessible to the authors.
9	White papers.

TABLE III
CLASSIFICATION SCHEME BASED ON RESEARCH QUESTIONS

RQ#	Categories	Properties/Attributes
1	Contribution Facets	Method/Technique/Approach, Framework, Model, Tool, Architecture, Platform, Empirical Study, Analysis, Comparison, Overview, or Others
2	Research Facets	Solution Proposal, Validation Research, Evaluation Research, Experience Papers or Others
3	Annual Popularity	Years between 2010 and 2015
4	Form of Experimentations	Stream only, Batch Only, Both
5	Contribution purposes	Usage enhancement, Performance enhancement, or Both
6	Data ingestion tool/sources	Kafka, Rabbit, ZeroMQ, Network socket, Twitter Streaming API, Kestrel, HDFS, Generic, Others
7	Number of nodes in experimentation	1-5, 5-20, or 20+
8	Data type(s) used in experimentation	Sensor, Social media, Graphical, Geospatial, Log, Raw, Generic, Web content, or Others

research facet have the better half of the rest. This indicates that contributors intend to validate their work in laboratories and even propose strong case studies about their contributions.

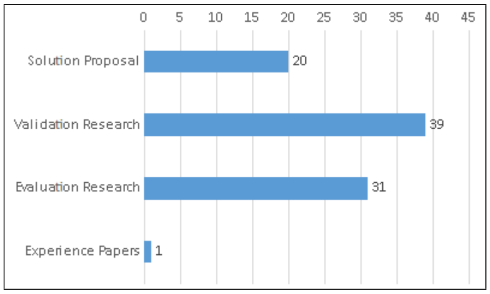

Fig. 2. Research Facet of the papers

RQ 3. What type of research methods are used for each of the framework in the papers?

Figure 3 illustrates the distribution of the frameworks in aspects of their contributions that are whether properly evaluated or just proposed as solutions. As it is shown in the figure, high portion of the contributions in each framework except InfoSphere is at least validated with either experiments, simulations or by prototyping. Also, the most of contributions about Flink are rather studied with case studies and the conclusions are supported in the papers. However, an opposite case appears for InfoSphere where the highest portion of the contributions appears as proposing solutions. There is also one experience research [18] exist in this study which is related to InfoSphere.

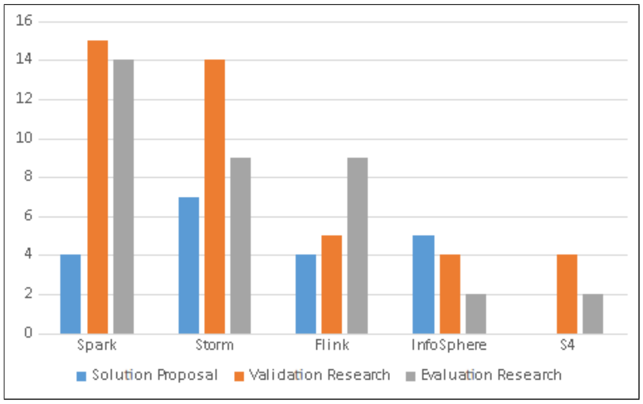

Fig. 3. The Research facets of each Stream Processing framework in the papers

RQ 4. What is the annual number of publications for each Big Data stream processing framework?

A study was conducted to see the annual popularity of each Big Data streaming frameworks. As it can be seen from Fig. 4, in 2015, Apache Spark, Storm and S4 have their highest number of publications, respectively 15, 14,and 2. This is while Apache Flink's highest publication quantity appears in 2013 with the count of 6 and IBM's InfoSphere has its highest quantity in 2014 with 4.

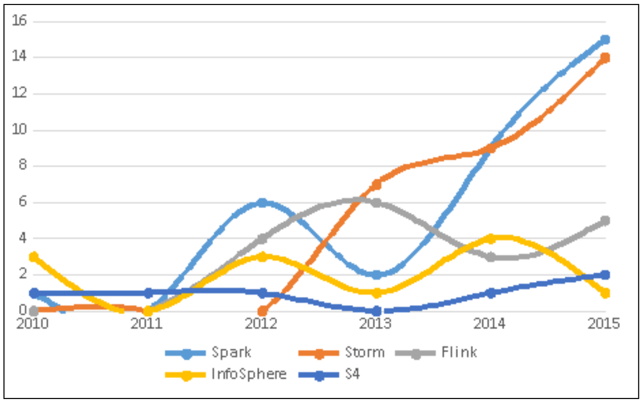

Fig. 4. Annual publication count for each the of frameworks

RQ 5. What is the ratio of experimentation type (batch only, stream only or both) used for each Big Data stream processing framework in the papers?

We analyzed experimentation types (Batch only, stream only or both) used for each Big Data stream processing framework in the papers. This shows the trends of researchers for in terms of experimenting preferences. As it can be seen from Fig. 5; the majority of the articles are based on stream processing, however, the quantity for Apache Spark's experimentation type almost equals both batch-only and stream-only jobs, respectively 16 and 18. The appeared trend of Spark here might indicate that Spark is a good preference for batch jobs as well. Also, the highest number of stream-based experimentation appears in Apache Storm with 23 times, while the least number appears in Apache Flink with 2 times which illustrates itself as a gap. This may be caused because of Apache Flink has been developed from a project called Stratosphere whose main goal is not stream-processing.

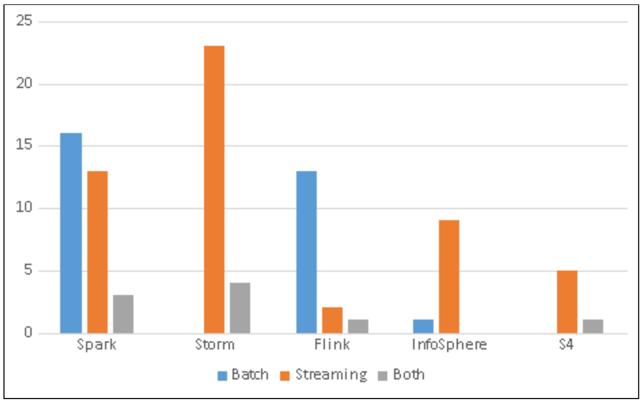

Fig. 5. Experimentation forms used in the papers for each framework

RQ 6. What is the ratio of contribution purposes (usage enhancement, performance enhancement or both) for each Big Data stream processing framework in the papers?

We conducted a cross analysis of the purpose of the contributions (usage enhancement, performance enhancement, or both) in the papers for each Big Data streaming framework to get a better understanding of the researchers to see their trends in the terms of what they aim with their contributions. Here, by ”usage enhancement” we refer to the articles that considers different use cases which result in enhanced proposals. Also, ”performance enchancement” refers to the articles that aims to improve the frameworks performance by either modifying the components of the frameworks or implementing new components. It can be seen from Fig. 6, IBM's InfoSphere has no performance-enhancement related contribution. In fact, IBM publishes regularly white papers about their products to explore technical aspects. However, white papers are excluded from this study. Also, the most contributions for almost each of the frameworks except Apache S4 is focused on further enhancing the usage of the framework itself.

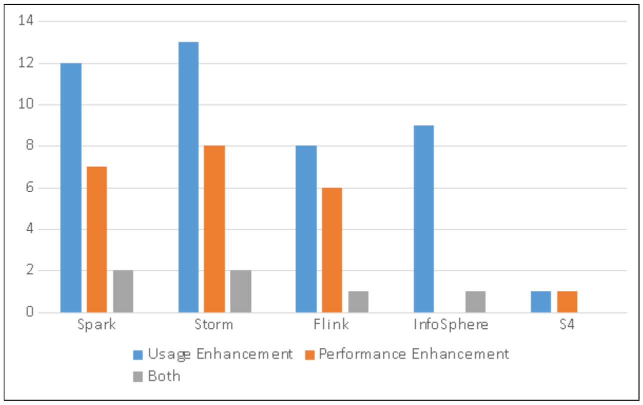

Fig. 6. Contribution purposes for each framework

RQ 7. Which data ingestion source is used most for each framework?

We conducted an analysis on the papers whose experiment types are based on streaming jobs. There are 55 papers proposing stream-based experimentation and our analysis includes looking into these 55 papers to specify the data-ingestion tools used in the experimentation phase within those papers. Most of the papers specify that the internal tools of the frameworks are used to ingest data from external sources, for example, [19] describes data ingestion in this way ”The MultimediaSpout” was in charge of retrieving the images from the external source, generating a stream of tuples, and passing it to the bolts”. As a result, the authors referred to these methods as ”Generic”. Also the papers about S4 platform do not mention any data ingestion methods except ”Generic” types. Hence, Fig. 4 was made by excluding the ”Generic” method and S4 platform. According to Fig. 7 Kafka and ZeroMQ/0MQ are the most widely used third party tool to ingest data from external sources also used by Apache Storm 5 times and 4 times, respectively. Whereas Apache Spark has ingested data 4 times from Kafka and 3 times from ZeroMQ/0MQ. Also, IBM's InfoSphere has used network sockets only beside the Generic method. The use of RabbitMQ and Kestrel appears with only Apache Storm.

RQ 8. What is the most preferred range for the number of nodes used in experimentation for each Big Data stream processing framework?

We conducted another cross analysis over the number of

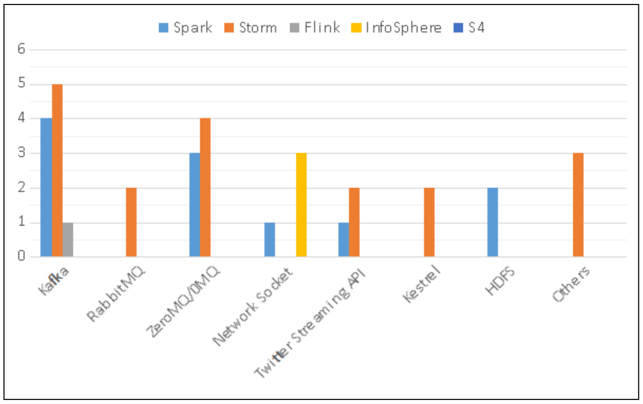

Fig. 7. Data ingestion tools/sources used for each framework

nodes used for each Big Data streaming framework in the experimentation phase of each paper. As Fig. 8 shows, papers about Apache Spark have a clear distinction from other frameworks in using of 5-20 nodes and 20+ nodes for experiments. Additionally, Apache Storm users mostly intended to use 1-5 nodes.

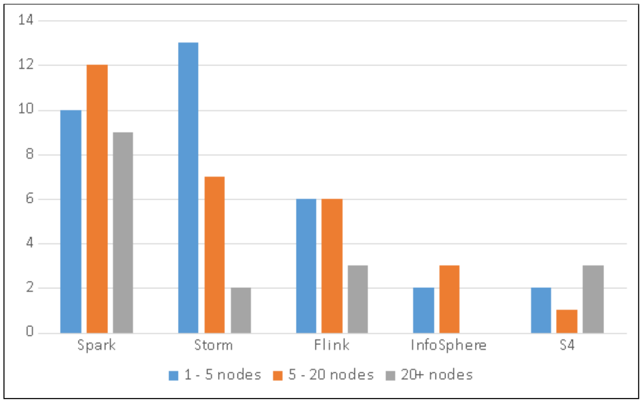

Fig. 8. Number of nodes used in experimentations for each framework

RQ 9. What type(s) of data is used most for each Big Data stream processing framework?

We conducted an analysis to see which data types are more preferred within the experimentation of the frameworks. For this analysis we excluded papers that uses datasets without giving some information about it or calling it ”dataset” directly, which we call them as ”Generic” data (20 papers) and the papers using raw data (30 papers) such as *txt* and *csv* files are included into exclusion also. As it can be seen from Fig. 9.; the most used data type is social media by all the frameworks. On the other hand, sensor data is the least attracted data type by the contributors. Considering ”Internet of Things” (IoT) as an important application area of Big Data, sensor data does not seem to get adequate interest by the researchers which illustrates itself as a gap in the field.

VI. Conclusion

The main objective of this study is to find trends and gaps about several SPFs for Big Data by using the Systematic Mapping approach. We present our findings which are also used as the starting point of a thesis study conducted by one of the authors in which some of the SPFs are being compared according to certain performance and requirements parameters.

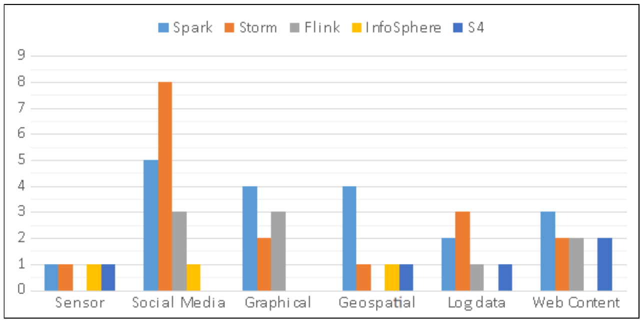

Fig. 9. Data types used by each of the frameworks in the experimentations

First of all, the first two RQs give overall information about interest of the community with the SPFs. The first RQ is about contribution facet of the articles. Findings to this RQ indicates that contributions are mostly based on proposing a new method, technique or approach to use with any of the SPFs (see Fig. 1), rather than modifying the architecture or model of the them. This case might imply that SPFs are adequately advanced, and the community is using them to solve different problems. Second RQ is about research facet of the articles. Answer to this RQ indicates that the most of contributors validate their study with at least by some kind of experiments rather than only proposing ideas as solutions to problems (see Fig. 2).

Third RQ takes a deeper look into the area by categorizing the research facets to each SPFs. The purpose of this analysis is to see the contributors with which SPF proposes the most evaluated researches and solutions without any validations. According to findings (see Fig. 3), Flink has its the highest portion with evaluated research. Spark and Storm have their the highest portion with validated researches. However, the highest portion of the InfoSphere related contributions denotes itself as solution proposals.

The data collected in this study highlights that Apache Spark and Storm have increasing number of publications each year whereas Apache S4 does not (see Fig. 4). This might be caused because S4 was developed by Yahoo! and then contributed to Apache software foundation. Consequently, Yahoo focused its attention to Apache Storm [6].

From the data collected, we could see that the experimentations in the papers related to Apache Storm, S4 and IBM's InfoSphere are mostly based on streaming jobs, where Apache Spark is almost equally experimented with both stream and batch jobs (see Fig. 5). The appeared trend of Spark here might indicate that Spark is a good preference for batch jobs as well. Additionally, Apache Flink is experimented with batch jobs in almost all of the papers which may reveals another gap.

Also, results depict that the researchers are mostly studied on the improvement of algorithms or optimization of tools for various use of frameworks, such as [23] and [24].

When we consider the data ingestion in Big Data streaming frameworks from external resources, we realized that most of the researchers used the frameworks' internal tools. However, Apache Spark and Apache Storm use Kafka, ZeroMQ/0MQ and Twitter's Streaming API for different reasons. This indicates that there are a numerous of alternative ways to stream data instead of using internal modules in the frameworks. Several gaps appear in the area of data ingestion methods within the frameworks (see Fig. 7). First one is the contributions about IBMs infoSphere use only networks sockets to retrieve data. For future works, we propose to use the alternative message queueing tools such as RabbitMQ and Kestrel for data ingestion purpose.

Size of cluster is an important aspect in the Big Data streaming framework area. From the data collected, it is observed that researchers mostly go either with up to 5 nodes or up to 20 nodes in their experimentations (see Fig. 8). Even though, there are a considerable number of papers that goes for up to 100 nodes for their experimentation, there is only one publication that uses more than 100 nodes [25].

Social media data is the mostly preferred source in consideration of the data types used in the experimentations (see Fig. 9). However, sensor data has the least attraction by the contributors which stays as a gap considering the important part of IoT technology within Big Data environment.

As a result, this SM study states several gaps and trends in the Big Data stream processing area which can help researchers to obtain an overview of the field and identify areas that require more attention from the research community.

References

[1] White, T., ”Hadoop: The definitive guide”, *O'Reilly Media, Inc.*, 2012.

[2] Apache Flink homepage. [Online]. Available: https://flink.apache.org/

[3] Apache Spark homepage. [Online]. Available: http://spark.apache.org/

[4] Apache Storm homepage. [Online]. Available: http://storm.apache.org/

[5] Apache S4 homepage. [Online]. Available: http://incubator.apache.org/s4/

[6] Quora: Is Yahoo going to open source its S4 ”Real-Time MapReduce” project? [Online]. Available: https://www.quora.com/Is-Yahoo-going-to-open-source-its-S4-Real-Time-MapReduce-project

[7] Grant, M. J., Booth, A., ”A typology of reviews: an analysis of 14 review types and associated methodologies” *Health Information and Libraries Journal*, 26(2), 91-108, 2009.

[8] IBM's InfoSphere Streams homepage. [Online]. Available: http://www-03.ibm.com/software/products/en/ibm-streams

[9] Morais, T.S.,”Survey on Frameworks for Distributed Computing: Hadoop, Spark and Storm” in *Proceedings of the 10th Doctoral Symposium in Informatics Engineering*,2015.

[10] Namiot, D., ”On Big Data Stream Processing”, *International Journal of Open Information Technologies*, vol.3, no. 8, pp. 48-51, 2015.

[11] Liu, X., Iftikhar, N., Xie, X., ”Survey of Real-time Processing Systems for Big Data”, in *Proceedings of the 18th International Database Engineering and Applications Symposium*, 2014, pp. 356-361.

[12] Osman, A., El-Refaey, M., Elnaggar, A., ”Towards Real-Time Analytics in the Cloud”, in *IEEE Ninth World Congress on Services*, 2013, pp. 428-435.

[13] Stonebraker, M., etintemel, U., Stan Zdonik, S., ”The 8 Requirements of Real-Time Stream Processing”, ACM's Special Interest Group on Management Of Data, vol. 34, no. 4, pp.42-47, 2005.

[14] Hu, H., Wen, Y., Li, X.,CHUA, T., ”Toward Scalable Systems for Big Data Analytics: A Technology Tutorial”, *IEEE Access*, vol. 2, pp. 652-687, 2014.

[15] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., ”Systematic Mapping Studies in Software Engineering,” in *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, 2008, pp. 68-77.

[16] Wieringa, R., Maiden, N., Mead, N., Rolland, C.. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, in *Requirements Engineering*, 2006, 11(1), 102-107.

[17] Solaimani, M., Iftekhar, M., Khan, L., Thuraisingham, B., Ingram, J., ”Spark-based anomaly detection over multi-source VMware performance data in real-time”, in *Computational Intelligence in Cyber Security (CICS)*, 2014 IEEE Symposium on (pp. 1-8). IEEE.

[18] Bouillet, E., Kothari, R., Kumar, V., Mignet, L., Nathan, S., Ranganathan, A., Turaga, D.S., Udrea, O. and Verscheure, O., ”Experience report: Processing 6 billion CDRs/day: from research to production”, in *Proc. 6th ACM Internat. Conf. on Distrib. Event-Based Syst.(DEBS12)* (Berlin, Ger., 2012) (pp. 264-267).

[19] Mera, D., Batko, M., Zezula, P., ”Towards Fast Multimedia Feature Extraction Hadoop or Storm,” in *IEEE International Symposium on Multimedia*, 2014, pp. 106-109.

[20] Akidau, T., Balikov, A., Bekiroglu, K., Slava Chernyak, S., Haberman, J., Lax, R., McVeety, S., Mills, D., Nordstrom, P., Sam Whittle, MillWheel, ”Fault-Tolerant Stream Processing at Internet Scale,” in *The Proceedings of the VLDB Endowment*, 2013, vol. 6, no. 11, pp. 1033-1044.

[21] Lin, L., Yu, X., Koudas, N., Pollux, ”Towards Scalable Distributed Real-time Search on Microblogs,” in *Proceedings of the 16th International Conference on Extending Database Technology*, 2013, pp. 335-346.

[22] Cherniack, M., Balakrishnan, H., Balazinska, M., Carney, D, etintemel, U., Xing, Y., Zdonik, S., ”Scalable Distributed Stream Processing,” 2015, in *International Conference on Management of Data*, pp. 811-825.

[23] Qiu, R.G., Wang, K., Shan Li, Dong, J., Xie, M., ”Big Data Technologies in Support of Real Time Capturing and Understanding of Electric Vehicle Customers Dynamics,” in *5th IEEE International Conference on Software Engineering and Service Science*, 2014, pp. 263-267.

[24] Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., Koutsopoulos, H., Moran, C., ”IBM InfoSphere Streams for Scalable, Real-Time, Intelligent Transportation Services,” in *ACM SIGMOD International Conference on Management of Data*, 2010, pp. 1093-1104.

[25] Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J.M., Ramasamy, K., Taneja, S., ”Twitter Heron Stream Processing at Scale,” in *ACM SIGMOD International Conference on Management of Data*, 2015, pp. 239-250.

Mohammed Alayyoub is a self-employed software engineer. He received BS degree in Software Engineering from Jordan University of Science and Technology (JUST), Jordan. He has completed his master's thesis in Software Engineering from Atilim University, Ankara, Turkey. His research interests include Big Data Stream Processing, Distributed and Parallel Processing.

Ali Yazici is a Professor and the Head of the Software Engineering Department at Atilim University, Ankara, Turkey. He received BS, and MS degrees in Mathematics from the Middle East Technical University (METU), Ankara, Turkey. He has completed his PhD dissertation at the Computer Science Department, Waterloo University, Canada. His research interests include Parallel Computing, Big Data and Cloud Computing, Database Systems, and e-Topics. He is the writer of many scientific articles, books and research reports in the field of Computing and Informatics. Dr. Yazici is a founding member of Turkish Mathematics Foundation, an associate member of Turkish Informatics Society, and a founding member of Turkish Informatics Foundation.

Ziya Karakaya is an Assistant Professor in the Computer Engineering Department at Atilim University, Ankara, Turkey. He received BS degree in Mathematics, and MS degree in Computer Education and Instructional Technologies, both from the Middle East Technical University. He has completed his PhD at the Modeling of Engineering Systems (MODES) with the main research area of Computer Engineering, at Atilim University. His research interests include Big Data, Cloud Computing, Parallel Computing, Distributed Computing, Virtualization, Object Oriented Software Engineering and Internet Programming. He is one of the authors of the book about Online Learning and Assessment (in Turkish). He has supervised many MS thesis and has many scientific writings, either published as Journal papers or as conference proceedings.