UNIVERSITY OF CAMBRIDGE
Cambridge Service Alliance

Big-Data Analytics Architecture for Businesses: a comprehensive review on new open-source big-data tools

Mert Onuralp Gökalp, Kerem Kayabay, Mohamed Zaki, Altan Koçyiğit, P. Erhan Eren, and Andy Neely

This is a Working Paper

Why this paper might be of interest to Alliance Partners:

Organisations suffer from a comprehensive architecture to manage and monitor the development of existing and new open-source big-data tools that are constantly growing. To determine the shortcomings and strengths of developing a big-data architecture with open-source tools from technical and managerial perspectives, this study (1) investigates the available open-source big-data technologies to present a comprehensive picture (2) presents a systematic method to review available open-source big-data tools (3) proposes an open-source reference architecture for big-data analytics (4) using the proposed architecture, analyses 15 firms to give directions to businesses for defining a big-data analytics architecture with the open-source big-data tools.

October 2017

Find out more about the Cambridge Service Alliance:
Linkedin Group: Cambridge Service Alliance
www.cambridgeservicealliance.org

Big-Data Analytics Architecture for Businesses: a comprehensive review on new open-source big-data tools

Mert Onuralp Gökalp[a] , Kerem Kayabay[a], Mohamed Zaki[b] , Altan Koçyiğit[a] , P. Erhan Eren[a] , and Andy Neely[b]

[a]Middle East Technical University, Informatics Institute 06800, Ankara, Turkey

[b]University of Cambridge, Institute for Manufacturing, CB30FS, Cambridgeshire, United Kingdom

Organisations suffer from a comprehensive architecture to manage and monitor the development of existing and new open-source big-data tools that are constantly growing. To determine the shortcomings and strengths of developing a big-data architecture with open-source tools from technical and managerial perspectives, this study (1) investigates the available open-source big-data technologies to present a comprehensive picture (2) presents a systematic method to review available open-source big-data tools (3) proposes an open-source reference architecture for big-data analytics (4) using the proposed architecture, analyses 15 firms to give directions to businesses for defining a big-data analytics architecture with the open-source big-data tools.

Introduction

In the big-data era, the information technology (IT) industry is continuously coming up with new models and distributed architecture to handle the exponentially increasing amount of data. Effectively integrating these models into its business processes, an enterprise can seek strategic advantage in the marketplace. Existing and emerging models do not just target the volume characteristic of big data. To exemplify, the speed at which data is processed is critical for timely decision-making and process-optimization activities. Some tools only aim to analyse data produced in a certain domain; for example, the Internet of Things. Many open-source and commercial tools continue to emerge to deal with the different characteristics of big data. As a result, there is an abundance of tools and platforms to analyse big data or act as building-blocks.

The tools used in the major functions of enterprises such as procurement, production, human resources, customer relations and marketing are seeking meaningful and up-to-date data. Digital service platforms using these tools make service exchange more efficient and effective [1]. As recent technologies and approaches such as Industry 4.0 are introduced, the characteristics of the data that these tools need to use also change radically over time. In the last decade, different platforms have been developed to address the needs of handling these diverse sets of characteristics of big data. These platforms are especially supported by companies that mainly operate on the Internet, such as Google, Yahoo, Twitter and LinkedIn. These companies aim to provide better services for their users and their third-party customers to increase their revenues by processing data generated by users in the Internet environment. They develop and utilize big-data analysis tools primarily to increase their ability to analyse, store and manage data from different heterogeneous sources. Most of the tools introduced have been made available to the IT industry as open-source tools, because building a solution on top of building an open-source platform enables companies to expedite their development process. Moreover, recent tools have provided more scalable and efficient solutions to process big data compared to traditional single-node solutions.

Extracting value from big data, an organization would train its IT workforce to obtain the technical expertise to be effective with those tools. Technical debt happens when businesses incur costs when they try to adopt these tools or change their existing source codes to run on newer versions [2]. As many platforms exist to process, store and analyse big data, an organization must choose the right set of tools to utilize as part of its data-analytics architecture. The choices depend on the characteristics of the data to be analysed, and the domain that the organization is operating under. There is no standard on how these tools fit together in a broader picture in the data-analytics architecture of an established organization. Most of these tools are unknown to the business world; yet this is a very active domain of research, and significant effort is focused on new open-source tools that are actively developed on the Apache Software Foundation (ASF)[1] and GitHub.[2]

Open-source tools have become the standard big data processing platforms [3]. Yet, in order not to lag behind the hype around big data, organizations outsource their big-data activities to commercial big-data solution providers. Commercial big-data solution providers typically rely on a subset of available open-source tools that may not fit the use case or organizational requirements of a firm. Therefore, outsourcing does not necessarily build the big-data capability of a firm, since it does not solve the technical, domain-specific and firm-specific soft challenges for establishing a big-data architecture. The abundance and sophistication of available open-source tools present a technical challenge that outsourcing solves, but the lack of domain-specific experience when using these tools presents another challenge. In the presence of technical and domain-specific competence, if data-driven culture and the right managerial skills do not exist within the firm, the value of the results from big-data analytics will remain underexplored.

There are studies in the literature that introduce frameworks and algorithms from a technical perspective [4–8]. There are also studies that systematically review the big-data domain from a managerial perspective [9,10]. There is a gap in the literature where open-source tools that exist for big data are systematically reviewed, explained and exemplified considering both the technical and managerial perspectives. Just by reviewing open-source tools, we have come across 241 tools in ASF and GitHub, and that number is final after applying strict filters, including reliability of the source, licence type, academic publication and last commitment

[1] Apache Software Foundation, https://www.apache.org, (accessed 4 October 2017).
[2] GitHub, https://github.com, (accessed 4 October 2017).

activity. There is no existing method to find and include emerging open-source tools addressing the components of the big-data analytics life cycle.

There are four main contributions of this paper. First, we systematically review the open-source tools and building-blocks of those tools that aim to store, manage and analyse big-data to present a comprehensive picture of the existing and improved technologies that will aid researchers to adjust their research directions. The second contribution is to provide a systematic method for tracking changes for the open-source tools, which is important since this is an active domain of research. There is a large array of open-source big-data tools available in the market, which are supported by communities of varying sizes and large corporations. We had to come up with a robust method while systematically reviewing all the open-source tools. The proposed method must filter all the prominent tools and be valid years after this publication has been published. Academia and technical personnel will be able to use the method introduced in this study to take the latest snapshot available before commencing a particular implementation or research. The third contribution is the open-source big-data analytics architecture, in which the introduced tools complement one another in a layered architecture to provide a comprehensive picture of the big-data analytics life cycle. The open-source big-data architecture provided simplifies building a unified and easier-to-implement big-data application for turning big-data opportunities into actionable and self-service data analytics. Fourth, we have reviewed the largest amount of case studies possible from secondary-data in order to clarify how large organizations utilize some of these tools as part of their business and decision-making processes. Instead of utilizing imposed solutions from commercial providers, businesses can utilize the proposed reference architecture and example case studies to devise their own big-data analytics architecture according to their use-case and organizational requirements.

This paper is organized as follows. In Section 2 related works about studies that focus on big-data tools are discussed. The method for systematic tool review is identified in Section 3. Section 4 presents the proposed open-source big-data tool stack and our findings. The managerial implications and development problems of big-data architecture are discussed in Section 5. Finally, we conclude the paper in Section 6.

1. **Literature Review**

The significant amount of data generated by a diverse and large number of data sources, including information services, IoT devices, social media and mobile devices, is not only too voluminous but also too fast and complex to be processed and stored using traditional methods. This exponential growth in data drives the industry and attracts researchers to develop new models and scalable tools to handle big data. This section reviews the studies that focus on investigating big-data tools and proposes a novel solution for big-data analytics in the literature.

In the literature there are numerous studies [6,11] that review and compare the existing popular big-data tool stacks, Apache Hadoop Stack [12] and Berkeley Data Analytics Stack

(BDAS) [13], to present the advantages and disadvantages of the tools in these stacks. Apache Hadoop stands out as a well-known open-source framework for big-data analytics. It is designed to work seamlessly with a stack of open-source tools to enable the storage and processing of a significant amount of data using clusters of commodity hardware. The Hadoop Stack includes a distributed file system, cluster management, storage, distributed processing and programming, data analysis, data governance and data pre-processing tools. Another important big-data tool stack is proposed by the AMPLab at the University of California, Berkeley, namely, the BDAS, which integrates open-source big-data tools to make sense of big data. It also includes a distributed file system, cluster management, distributed data processing and programming, data analysis, data pr tools and in-house developed big applications. Another study [4] reviews the current state-of-the-art in open-source big-data analytics tools for machine learning and provides recommendations for evaluating these tools. However, these studies review existing big-tool stacks and do not provide a comprehensive review of all the available big-data analytics solutions which this study aims to contribute.

Other studies propose [7,14] a high-performance computing stack for big-data analytics and summarize the capabilities of these stacks in 21 identified architecture layers. They review over 300 software packages to define this tool stack. The recent studies [8,10] review the literature systematically to analyse the current state and research directions of big data. The study of Grover et al. [8] moves beyond the systematic literature review and presents a limited number of conventional big-data tools to provide an overview thereof. A recent survey [5] provides a global view of state-of-the-art big-data technologies and compares these technologies in different system layers. This study mainly focuses on discussing the Hadoop framework and tools developed on top of it, and commercial Hadoop distributions such as Cloudera,[3] Hortonworks,[4] MapR,[5] and so on. Similarly, a recent study [15] gives an overview of widely used big-data technologies to identify the key features of these technologies. However, these studies do not provide any information about how to review these tools systematically, the nature of the criteria for including a tool in this stack or how to bring these tools together to define a big-data architecture. In this study, we propose a systematic method to review open-source tools and give directions about how to select a big-data tool for their big-data use-cases.

There are also some studies [16–19] that propose a reference architecture for big-data analytics. One study [16] presents a reference architecture for semantically aware big-data systems by taking into account the unique characteristics of big data. Another study by Pääkkönen et al. [17] proposes a technology-independent reference architecture for big-data systems. The authors also classify the related commercial big-data technologies and products based on analysis of the published use-cases. There are some domain-specific solutions to present a reference architecture for big-data analytics. The study of Geerdink [18] proposes a

[3] Cloudera, https://www.cloudera.com/ (accessed 26 September 2017).
[4] Hortonworks, https://hortonworks.com/ (accessed 26 September 2017).
[5] MapR, https://mapr.com/ (accessed 26 September 2017).

reference architecture to guide software architects, mainly in defining big-data analytics solutions for predictive analytics using qualitative data analysis and evaluated using a questionnaire that investigated several quality criteria. Another study [19] focuses on presenting a reference big-data analytics architecture for typical national defence domain requirements. Moreover, the authors demonstrate how to use the proposed reference architecture to define concrete architecture for a use-case. Nevertheless, these studies mainly focus on proposing use-case-specific architectural solutions from a technical perspective. They do not take into account the managerial perspective about how to assess a big-data tool for different requirements, or the current state of big-data tools to capture value for future projects. Moreover, they mention some conventional big-data tools to illustrate the applicability of their architecture, but they do not provide a comprehensive review of existing open-source big-data tools.

2. **Research Method**

One of the main contributions of this study is to develop a systematic approach to seek out existing open-source big-data tools in the market. To this end, a systematic tool-review method was used based on the following three-phased systematic literature review approaches, as described by Tranfield et al. [20] and Kitchenham and Charters [21,22].

- *Phase-I* includes the planning of the review process and developing the review protocol according to the research aim and objectives.
- *Phase-II* conducts the review process to identify, select and evaluate the open-source tools.
- *Phase-III* comprises a basis for examining research results and reporting them with qualitative and quantitative results.

In this section, the activities in Phase-I and Phase-II are explained in detail. The results obtained in this systematic review process for Phase-III are synthesized and presented as a comprehensive analysis in the form of open-source big-data reference architecture in Section 4.

3.1 Phase-I: Planning the review process

The first phase of the systematic tool-review process is to develop a protocol that includes defining the objectives and research questions of the review process to form a basis for finding proper databases to search open-source tools, developing a research strategy, as well as synthesis of the method.

The main objective of this study is to investigate the available and trending open-source big-data tools to determine the shortcomings and strengths of developing a big-data architecture from technical and managerial perspectives. According to our findings, there is a large array of open-source big-data tools available in the market; however, there is no study in the literature that provides a comprehensive picture, especially for big-data tools, to comprise researchers, small and medium-sized enterprises (SME), established firms, commercial big-data solution providers and software architects and developers. To this end, this systematic tool-review process aims to provide the state-of-the-art for open-source big-data tools to explain what type of tools are missing and mature enough for researchers to adjust their research directions in the big-data domain. This will help firms to develop their big-data development strategies to assess what type of tool fits their needs and use-cases, and what type of capabilities they lack to perform efficient data processing in their solution domain. Moreover, this study also aims to assist software architects and developers with which tools are ready to use for their big-data applications from an open-source perspective. According to these objectives, two main research questions (RQ) were determined, by which the review process was driven, as follows:

- *RQ-1:* What are the major categories of open-source technologies employed to overcome big-data challenges?
- *RQ*-2: What are the contributions of the technology companies to open-source big-data tools?
- Based on the importance of evaluating proper tools in determining the overall validity of the tool-review process, several suitability conditions, including inclusion as well as exclusion criteria, are defined. In the literature, there is no such attempt to define a systematic review process protocol for open-source tools. Therefore, a novel review protocol for open-source tools was defined based on the defined research questions and objectives.
- *Condition-I:* The review was conducted by searching the Apache Software Foundation (ASF) projects and GitHub database. The GitHub database includes more than 19.4 million open-source projects [23]. Moreover, the project information and source codes can be gathered easily through the provided application programming interface (API), which enables us to automatize the review process to reduce researcher bias. In addition to ASF projects and the GitHub repository database, Google Open Source,[6] Facebook Open Source[7] or IBM Open Source[8] platforms also provide alternative databases for their open-source projects. Nevertheless, the projects included in these platforms are already maintained in GitHub repositories. To collect data about the open-source big-data tools, the ASF projects and GitHub database were searched using the following keywords:

big?data?analytics OR big?data?analysis OR stream?processing OR batch?processing OR real?time?processing OR complex?event?processing OR distributed?messaging OR distributed?file?system OR map?reduce OR distributed?resource?management OR distributed?database OR scalable?machine?learning

- *Condition-II:* In GitHub, starring a project repository enables users to keep track of these projects and demonstrate users' interest in the project. Therefore, as a quality

[6] Google Open Source, https://opensource.google.com/ (accessed 6 October 2017).
[7] Facebook Open Source, https://code.facebook.com/projects/ (accessed 6 October 2017).
[8] IBM Open Source, https://developer.ibm.com/code/open/ (accessed 6 October 2017).

control of the tools and review process, only those tools were selected that have at least 100 stars in the GitHub database.

- *Condition-III:* The data about open-source tools was collected from the GitHub repositories, official web page (if any), as well as their documentation (if any), mailing lists and forums. The data collected was inspected to clarify whether or not the tool provides a solution in the big-data domain. The key difference between traditional data processing and big-data processing lies in how the processing is executed. Traditional data analytics can be performed on a stand-alone basis. However, in big-data analytics, the data and processing capabilities should be broken down and executed across multiple nodes. Therefore, the inevitable characteristic of a big-data analytics tool is scalability by supporting distributed processing and storage. Moreover, the big-data tools should be readily available and continue operating properly in the event of the failure of some of these distributed nodes to build a reliable big-data infrastructure.
- *Condition-IV:* In order to specify active projects, only tools were selected whereby there has been a commitment to the source code in the last six months (in our case after 1 March 2017).
- *Condition-V:* The acquired big-data tools were also evaluated by having a credible open-source licence, which defines how the source code of the software can be modified or distributed. Moreover, most of the credible licensed open-source projects include strong user communities via forums and/or mailing lists for support, training and consultation purposes. There exist many open-source licences acknowledged by the Open Source Initiative (OSI).[9] The most popular open-source licences are Apache Licence 2.0, Berkeley Software Distribution (BSD), GNU General Public License (GPL), GNU Lesser General Public Licence (LGPL), MIT Licence, Mozilla Public License (MPL) and Eclipse Public License (EPL).

The tools that do not have any credible licence were also evaluated according to the inclusion condition in the following:

- *Condition-VI:* Tools with at least one academic publication that demonstrates applicability and benefits to the big-data domain were included in this study.

The overall conditional review process can be denoted mathematically as follows:

$$Condition\ I\ and\ Condition\ II\ and\ Condition\ III\ and\ Condition\ IV\ and\ (Condition\ V\ or\ Condition\ VI)$$

3.2 Phase-II: Conducting the review process

This sub-section explains in detail the activities of conducting the review process for responding to the research questions, and demonstrates the outcomes of the review process with both qualitative and quantitative results by examining the relevant open-source big-data tools throughout the ASF projects and GitHub database.

[9] Open Source Initiative, https://opensource.org/ (accessed 27 September 2017).

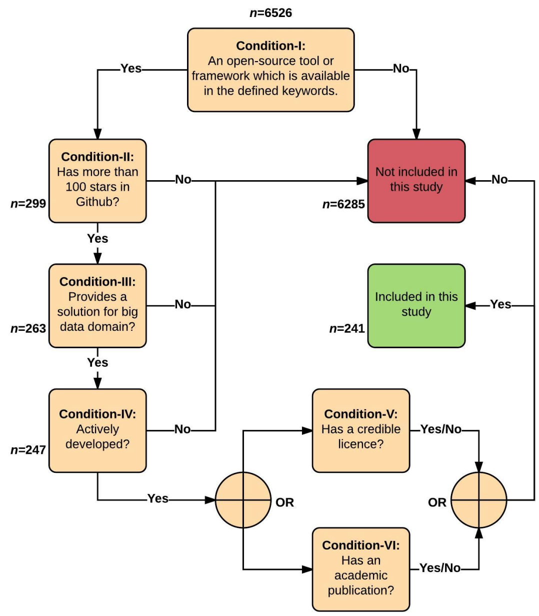

Figure 1. The Flowchart of the Review Process

In our previous study [24], we discussed the applicability of web crawling to discover relevant information by leveraging the latest advancements in distributed computing and big-data analytics technologies. With the help of knowledge gained from this study, we have attempted to collect data about open-source big-data-specific tools from the ASF projects and GitHub database using separated automatized crawlers for ASF and GitHub. The implemented script for the ASF project database basically traverses all of the ASF projects, searches defined keywords in their official web page and returns the names, as well as links, of the available project as an output. The other script for crawling the GitHub projects database utilizes GitHub search API to collect project names and links in defined keywords, and filters projects, respectively, according to Condition-II and Condition-IV. The rest of the conditions were assessed manually by researchers. There are many advantages to using an automatized approach in the systematic tool-review process. First of all, it enables us to find as many primary tools as possible that are related to the defined review objectives, and distinguishes systematic reviews from traditional research using an unbiased and transparent search strategy. The automatized approach also helps us to update our survey data about big-data tools, periodically and with no extra effort. Moreover, some of the tools may appear more than once for different defined keywords. In such a large tool data set, it is difficult to sift through the tools manually to reveal unique projects. Therefore, a programmatic approach is needed to deal with such a significant amount of data for the systematic tool-review process.

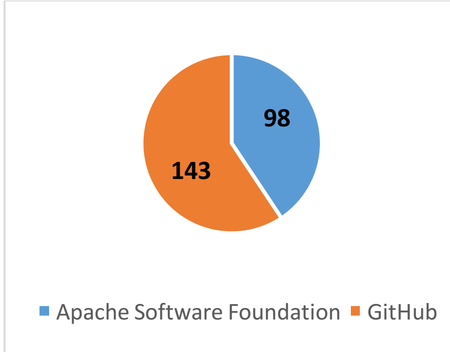

Figure 2. The Distribution of the Apache and GitHub Projects

The review process was completed in September 2017. The project that is found in the ASF projects database was also reviewed in the GitHub repositories to check that they ensure all of the conditions defined above. As a result of time constraints, the reviewed tools were not evaluated by configuring and running each of them. Through using the abovementioned list of keywords and automatized crawlers, initially 6,526 big-data tools were identified from the ASF project and GitHub database. After assessing these 6,526 big-data tools according to the conditions mentioned in Phase-I, 6,285 tools were discarded, and finally 241 open-source big-data tools were selected and taken forward to propose an open-source big-data reference architecture. The flowchart of the review process and results are depicted in Figure 1. This descriptive investigation is also depicted as Appendix-A, designed to contain the name of the tool, the source of the tool, the link of the main project page and the category of the tool for all 241 unique open-source big-data tools. Moreover, the distribution of the projects in GitHub and ASF is also depicted in Figure 2. A wide variety of open-source tools have been developed, and continue to be developed, to process, store, analyse, manipulate, aggregate, manage and visualize big data in accordance with our systematic tool-review process.

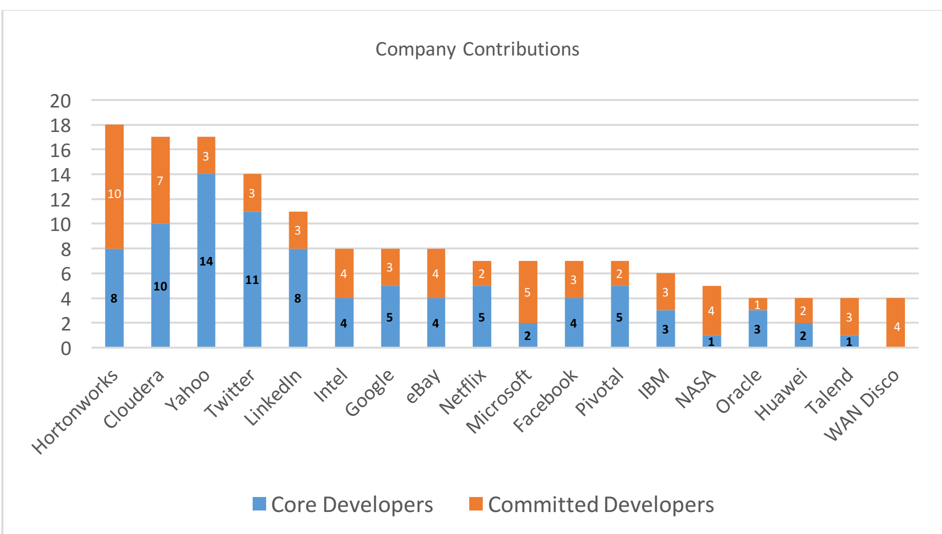

Figure 3. Open-Source Big-Data Tools Company Contributions

The contribution of the leading technology companies to open-source big-data tools was also investigated to clarify RQ-2. To this end, we followed different methodologies for the ASF and GitHub projects. For the ASF projects, we examined the initial proposals, which present the core and committed developers, and their affiliations, of each big-data tool. For the GitHub projects, the project definitions and the main repository in which the tool was developed were investigated. Hence, the distributions of the companies for the open-source big-data tools that are depicted in Figure 3 were obtained. The figure depicts only the companies that make contributions to more than three big-data tools in order to emphasize the contributions of these companies. We were able to distinguish core and committed developers with the help of a rich stream of information in the ASF project proposals and GitHub repositories. As can be seen from the figure, Hortonworks and Cloudera are the most active companies in open-source big-data tool development, because these companies already serve as big-data solution providers, especially for the Apache Hadoop. The well-known internet companies, Yahoo!, Twitter and LinkedIn follow these big data solution providers due to massive amount of batch and real-time data that they are handling. These giant companies are contributing to the development of big-data tools to provide a solution for their in-house problems and to develop a community around their tools to increase productivity. For example, Twitter recently open-sourced its project Heron[10] [25,26], a real-time stream-processing engine, in ASF. Twitter mainly utilizes Heron to process thousands of tweets every second with the purpose of detecting trend topics in real time. Heron has already attracted contributors from leading technology companies such as Google and Microsoft, which want to extend their real-time stream processing. Another important observation from the company distributions for open-source big-data tools is that the leading technology company, Google, is not at the top of the list. However, most of the open-source big-data tools are implemented based on their breakthrough publications: BigTable [27], MapReduce [28], MillWheel [29], Pregel [30] and FlumeJava [31].

3. **Open-Source Architecture for Big-Data Analytics**

Big-data applications require special methods, tools and techniques to handle data efficiently from the initial stage, to collect data to the final stage, and for value creation, as a result of the inherent characteristics of big data. Therefore, an efficient big-data application needs to be adapted to the characteristics of the data to be used, and it is necessary to utilize the appropriate tools for these properties. Therefore, in this section, RQ-1 is clarified by using the knowledge gained in the systematic tool-review process, an open-source architecture for big-data analytics is presented and explained in detail. The overall process of leveraging big data to drive decision-making can be broken down into two main processes: *data management* and *data analytics* [32–34]. The *data-management* process is responsible for acquiring, governing, integrating, securing and storing data to prepare it for applying data-analytics methods. *Data analytics*, on the other hand, deals with data modelling, analysing and interpretation to transform raw data into valuable knowledge. According to the process of sifting the results derived from our systematic tool review, the available open-source big-data

[10] Heron, https://twitter.github.io/heron/ (accessed 27 September 2017).

tools for data management fall into five different categories: distributed file system, cluster management, data store, governance and security and data ingestion. Data analytics, on the other hand, includes distributed data processing and programming, visualization, data analysis and data pre-processing components. The application and supporting tools components include the technologies that can be utilized for both of the big-data processes. A reference open-source big-data architecture for big-data analytics is proposed, which is depicted in Figure 4. Each component in the proposed reference architecture is optional according to the requirements and application domain. These 11 different components are placed in the reference architecture by taking into account inherent characteristics of these components and their interactions with one another. In the following, we briefly explain the components of the proposed reference architecture for big-data analytics. Thereby, the technical depth is beyond the scope of this study and an exhaustive list of the tools of each component is given in Appendix-A; the following explanations represent a relevant subset of the lesser-known open-source tools to help academia and industry in building a unified architecture for their different kinds of big-data use-cases, such as predictive analytics, social media analytics, text analytics, audio analytics and video analytics.

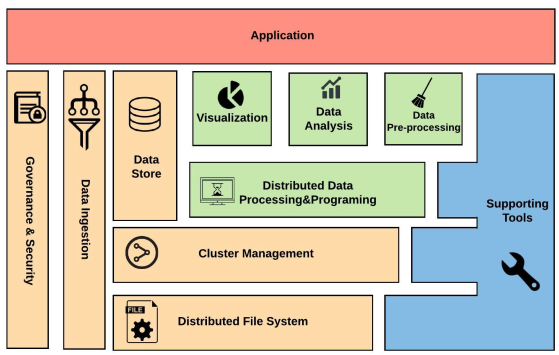

Figure 4. Proposed Open-Source Architecture for Big-Data Analytics

- **Distributed file system:** The distributed file system (DFS) layer resides at the lowest level of this architecture to store and manage large amounts of data across multiple nodes of commodity hardware. DFS is a basic file system that allows disks in a distributed environment to behave as a single virtual disk by breaking the data down into smaller pieces and distributing them throughout the cluster. DFS are commonly designed to conform with master and slave node architecture. Master nodes are responsible for managing job submissions by distributing jobs coequally to slave nodes, which manages processing and collects results from slave nodes. The main benefit of master–slave architecture is the ability to increase the number of slave nodes in the cluster to support vertical scalability. The well-known DFS is called the Hadoop Distributed File System (HDFS) [35] but there are also some other alternative tools, such as the Baidu File System,[11] the Gluster File System[12] and Alluxio,[13] formerly known as Tachyon [36,37].

[11] Baidu File System, https://github.com/baidu/bfs (accessed 27 September 2017).
[12] Gluster File System, https://www.gluster.org/ (accessed 27 September 2017).
[13] Alluxio, http://www.alluxio.org/ (accessed 27 September 2017).

- **Cluster management:** This architectural component is responsible for deployment, scheduling and orchestrating the jobs across the large networks of nodes to build a readily available and highly scalable computing infrastructure. Therefore, choosing a suitable cluster-management tool is vital for the overall performance of the big-data infrastructure. Apache Mesos[14] [38], Apache Aurora,[15] Genie-Netflix[16] and Apache Helix[17] can be listed as examples of open-source cluster-management tools for big-data analytics.
- **Distributed data processing & programming:** Big-data use-cases may need to process significant amounts of batch data or millions of data tuples in real time to build a data-analysis model and produce results in a timely manner. This significant amount of processing load cannot be handled using traditional methods with a single node solution. To this end, there is a need to constitute an efficient and scalable or distributed programming model and processing solutions, which should be able to deal with the volume and velocity characteristics of big data. The distributed data-processing tools vary within themselves; however, there are two notable processing methods: batch processing and stream processing. MapReduce [28] is the well-known programming model that implements parallel processing jobs for big-data sets. Distributed batch data-processing tools such as Apache Spark[18] [39] and Hadoop use the MapReduce programming model. On the other hand, Apache Storm[19] [40], Heron, Gora[20] and Apache Samza[21] can be listed as examples of assessed stream-processing tools.
- **Data Store:** The significant amount of data generated by the diverse and large number of data sources is not only too voluminous but also too fast and complex to be stored using traditional storage technologies. In an attempt to store this big data, distributed, scalable, schema-free and fault-tolerant big-data storage technologies that are compatible with a distributed file system are needed. These requirements trigger the development of NoSQL databases, which are increasingly being used in big-data applications. Several types of NoSQL database, which are column-based, key-value-based, document-based, graph-based and time-series-based, have been proposed to support specific needs and use-cases. In our systematic tool-review process, 37 different big-data storage technologies were founded. Tera,[22] RethinkDB,[23] HBase,[24] Voldemort[25] and RQLite[26] are some of these storage tools.
- **Visualization:** Visualization of the data is the ability to present a massive amount of data in a pictorial or graphical format to enable decision-makers to interpret difficult concepts or identify new patterns easily. As expected, big-data visualization techniques differ from traditional data-visualization approaches because of the unique characteristics of big

[14] Apache Mesos, http://mesos.apache.org/ (accessed 27 September 2017).
[15] Apache Aurora, http://aurora.apache.org/ (accessed 27 September 2017).
[16] Genie-Netflix, https://netflix.github.io/genie/ (accessed 27 September 2017).
[17] Apache Helix, http://helix.apache.org/ (accessed 27 September 2017).
[18] Apache Spark, https://spark.apache.org/ (accessed 27 September 2017).
[19] Apache Storm, http://storm.apache.org (accessed 1 October 2017).
[20] Apache Gora, http://gora.apache.org/ (accessed 27 September 2017).
[21] Apache Samza, http://samza.apache.org/ (accessed 27 September 2017).
[22] Tera, https://github.com/baidu/tera (accessed 27 September 2017).
[23] ReThinkDB, https://rethinkdb.com/ (accessed 27 September 2017).
[24] HBase, http://hbase.apache.org/ (accessed 27 September 2017).
[25] Voldemort, https://github.com/voldemort/voldemort/tree/master (accessed 27 September 2017).
[26] RQLite, https://github.com/rqlite/rqlite (accessed 27 September 2017).

data, such as displaying a high volume of data without collapsing/condensing, dealing with continuously flowing real-time data and separating a variety of categories and structures of data seamlessly. As a result of these challenges, there are a limited number of open-source big-data visualization tools available. Kibana[27] and Airpal[28] are some of the data-visualization tools.

- **Data analysis:** Data analysis is the process of developing an analytical model by examining raw data sets in order to infer knowledge by finding patterns and drawing conclusions with the aid of specialized tools and algorithms. In order to develop a successful data-analysis model, predictive modelling, querying, machine learning and deep learning are indispensable technologies. To this end, querying tools on distributed storage systems, machine learning and deep-learning libraries that support distributed processing are included under this architectural component. In the final stage of tool assessments, 50 different data-analytics tools remained. Among these tools are Apache Calcite,[29] Apache Drill,[30] Tensorflow[31] [41], PhotonML,[32] Cascalog[33] and Scalding.[34]
- **Data pre-processing:** Data scientists face many challenges regarding finding a reliable analysis method when dealing with big data. One of these challenges is data cleaning to detect noises, errors or incomplete data to improve the overall success of data analysis. Another important data pre-processing challenge is about the collection of data from outside sources and transforming these data sets to load in-house data storage systems to maximize the strength of data analytics. CKAN,[35] Apache Griffin[36] and Data Cleaner[37] are among the open-source big-data tools assessed in this category.
- **Governance & security:** As organizations adapt big-data analytics to capture nascent opportunities, data governance and data security can pose key challenges that may affect the entire big-data architecture. Moreover**,** the big-data applications tend to present specific governance and security policy enforcements for each individual use-case about the data they have collected. Therefore, this component of our reference architecture mainly addresses open-source solutions for data governance, data security, service programming and benchmarking. For example, Apache Atlas[38] provides a scalable and extensible set of core foundational governance services. Apache Ranger[39] proposes a data-security framework for monitoring and managing the security of data across the Hadoop platform. HiBench,[40] which was developed by Intel, is a big-data benchmark suite that helps to evaluate different big-data frameworks in terms of speed, throughput and

[27] Kibana, https://github.com/elastic/kibana (accessed 27 September 2017).
[28] Airpal, http://airbnb.io/airpal/ (accessed 27 September 2017).
[29] Apache Calcite, https://calcite.apache.org/ (accessed 27 September 2017).
[30] Apache Drill, http://drill.apache.org/ (accessed 27 September 2017).
[31] Tensorflow, https://www.tensorflow.org/ (accessed 27 September 2017).
[32] PhotonML, https://github.com/linkedin/photon-ml (accessed 27 September 2017).
[33] Cascalog, https://github.com/nathanmarz/cascalog (accessed 27 September 2017).
[34] Scalding, https://github.com/twitter/scalding (accessed 27 September 2017).
[35] CKAN, https://ckan.org/ (accessed 27 September 2017).
[36] Apache Griffin, http://griffin.incubator.apache.org/ (accessed 27 September 2017).
[37] Data Cleaner, https://github.com/datacleaner/DataCleaner (accessed 27 September 2017).
[38] Apache Atlas, http://atlas.apache.org/ (accessed 27 September 2017).
[39] Apache Ranger, http://ranger.apache.org/ (accessed 27 September 2017).
[40] HiBench, https://github.com/intel-hadoop/HiBench (accessed 27 September 2017).

system resource utilizations. Apache Zookeeper[41] is a service programming tool to develop and maintain extremely reliable distributed coordination across nodes.

- **Data ingestion:** Data ingestion tools help in transferring data from various outside data sources to internal systems in the most efficient and reliable way. They also provide a resilient and fault-tolerant data-distribution method across the architectural components. By taking into account the volume and velocity characteristics of big data, data-transferring tools play a crucial role, not only in importing data into big-data platforms but also in the overall performance of the big-data applications. One of the well-known data-ingestion tools is Apache Kafka[42][42], which is pioneered by LinkedIn. Sqoop,[43] Pulsar,[44] Gobblin[45] and Suro[46] are some of the data-ingestion tools that were discovered in the tool-review process.
- **Application:** This layer mainly provides high-level abstraction to implement specific big-data applications and/or present the analysis results produced by the underlying layers to end-users. For example, Nutch[47] [43] is a production-ready web crawler, which is also extremely extensible and scalable in the processing of big data. KillrWeather[48] is another reference application to integrate streaming and batch data processing with well-known open-source tools such as Apache Spark for distributed stream processing, Apache Cassandra[49] for data storage, Apache Kafka to ingest data and Akka[50] for service programming.
- **Supporting tools:** As a result of our tool review, there exist some specific open-source big-data tools that do not fit any other components in the proposed reference architecture, such as Apache Edgent[51] for edge-programming, which enables the implementation of applications for small footprint edge devices; Apache Knox[52] as an application gateway tool to provide a single access point for all REST and HTTP interactions; Apache Tephra[53] for transaction management to provide globally consistent transactions on top of distributed data stores; Apache OpenWhisk[54] for emerging serverless computing technology to execute big-data functions in response to events; Apache River[55] as a networking tool to define scalable and flexible network systems; and Apache Solr[56] as a search-server.

---

[41] Apache Zookeeper, http://zookeeper.apache.org/ (accessed 27 September 2017).
[42] Apache Kafka, https://kafka.apache.org/ (accessed 27 September 2017).
[43] Apache Sqoop, http://sqoop.apache.org/ (accessed 27 September 2017).
[44] Apache Pulsar, http://pulsar.apache.org/ (accessed 27 September 2017).
[45] Apache Gobblin, http://gobblin.incubator.apache.org/ (accessed 27 September 2017).
[46] Suro, https://github.com/Netflix/suro (accessed 27 September 2017).
[47] Apache Nutch, http://nutch.apache.org/ (accessed 27 September 2017).
[48] KillrWeather, https://github.com/killrweather/killrweather (accessed 27 September 2017).
[49] Apache Cassandra, http://cassandra.apache.org/ (accessed 27 September 2017).
[50] Akka, http://akka.io/ (accessed 27 September 2017).
[51] Apache Edgent, http://edgent.incubator.apache.org/ (accessed 27 September 2017).
[52] Apache Knox, http://knox.apache.org/ (accessed 27 September 2017).
[53] Apache Tephra, http://tephra.incubator.apache.org/ (accessed 27 September 2017).
[54] Apache OpenWhisk, http://openwhisk.incubator.apache.org/ (accessed 27 September 2017).
[55] Apache River, http://river.apache.org/ (accessed 27 September 2017).
[56] Apache Solr, http://lucene.apache.org/solr/ (accessed 27 September 2017).

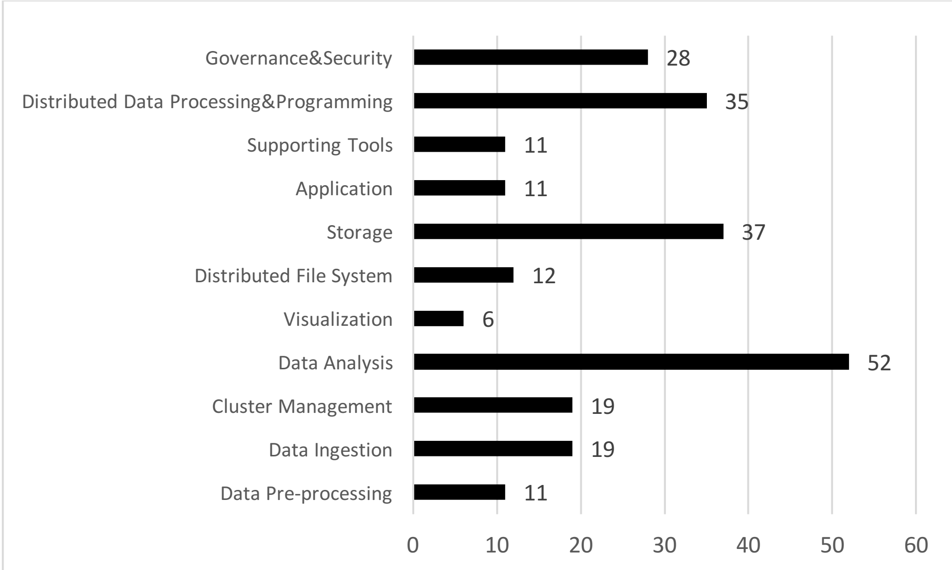

Figure 5. Distribution of Open-Source Big-Data Tools

Besides the well-known tool stacks such as the Apache Hadoop tool stack or BDAS, practitioners and academics can constitute their own solutions by unifying a suitable tool for each of the architectural components in the proposed reference architecture. For example, Streamlio[57] provides a unified solution for real-time streaming processing, with Apache Pulsar for data ingestion, Apache Heron for distributed data processing and programming, Apache Bookkeeper[58][44] for data governance, and Kubernete[59] for cluster management. All of the technologies in the solution domain of Streamlio are relatively new and developing technologies. The unified solution of Streamlio shows that practitioners and academics do not have to utilize a well-known tool stack for their big-data use-cases; a better solution can be unified by choosing the tools that best fit their requirements. To this end, we have explained in detail how to choose a big-data tool in the next sub-section.

3.1. How to choose a big-data tool?

We have depicted a comprehensive review of the tools for each architectural component in Figure 5 and in Appendix-A. As can easily be observed, there are plenty of tools for each architectural component. At this point, it is crucial to decide which tool is most suitable for the inherent characteristics and requirements of your big-data use-case to define a complete software architecture and to obtain the maximum benefit from this architecture. To this end, we have reviewed some secondary data-sets about big-data from industry, and academic studies from technical and managerial perspective to clarify important factors in big-data tool selection. We have collected 113 different secondary data, which includes real-world use cases, solution briefs, whitepapers as well as blog posts for big-data from a wide range of

[57] Streamlio, https://www.streaml.io (accessed 27 September 2017).
[58] Apache Bookkeeper, https://bookkeeper.apache.org (accessed 27 September 2017).
[59] Kubernete, https://github.com/kubernetes/kubernetes (accessed 27 September 2017).

industries such as; telecommunication, healthcare, banking & finance, manufacturing, transportation, energy and so on, from leading technology companies and big data solution providers as depicted in Table 1. As a result of reviewing process of this secondary data-set, we have come up with important criteria which are timing, data-size, platform independency, and data storage model to choose a particular big-data tool for different set of big-data application requirements. These secondary data-sets also help us to support the proposed big-data reference architecture by combining scholarly data and real-world data, as indicated in Hevner's information system research framework [45].

Table 1. Secondary Use-Case Company Distribution

Company Name	Number of Secondary-Data
Data Torrent[60]	5
Data Bricks[61]	11
Data Meer[62]	9
Facebook (Engineering Blog)	6
Hortonworks	8
Informatica[63]	2
MapR	8
Mesosphere[64]	5
Pentaho[65]	12
Pivotal[66]	15
Splunk[67]	2
Talend[68]	3
Teradata[69]	15
Twitter (Engineering Blog)	8
Yahoo (Engineering Blog)	4

Timing requirement: One of the most important tool-selection criteria for big-data applications is the timing requirement as also discussed in the academic studies [4,46]. The stream-processing tools are the most suitable when there is a need to respond immediately to certain events. On the other hand, batch-processing tools, which have loose timing

[60] Data Torrent, https://www.datatorrent.com/ (accessed 12 October 2017).
[61] Data Bricks, https://databricks.com/ (accessed 12 October 2017).
[62] Data Meer, https://www.datameer.com/ (accessed 12 October 2017).
[63] Informatica, https://www.informatica.com/ (accessed 12 October 2017).
[64] Mesosphere, https://mesosphere.com/ (accessed 12 October 2017).
[65] Pentaho, http://www.pentaho.com/ (accessed 12 October 2017).
[66] Pivotal, https://pivotal.io/ (accessed 12 October 2017).
[67] Splunk, https://www.splunk.com/ (accessed 12 October 2017).
[68] Talend, https://www.talend.com/ (accessed 12 October 2017).
[69] Teradata, http://www.teradata.com/ (accessed 12 October 2017).

requirements, are much more applicable for working with aggregated data to extract valuable knowledge. Moreover, the identification and inference of certain events, such as the detection of unexpected occurrences for a timely reaction, may require the incorporation of complex event-processing tools into big-data architecture. It is very crucial to choose a proper solution to address your big-data requirements in a production environment. The companies can improve their efficiency and decrease their operational and management cost by applying proper processing method in their big-data application. For example, in a case study provided by Pentaho [47], a healthcare company is addressing the changing healthcare environment by providing hospitals with real-time data to improve quality of care, efficiency and operations. They mentioned that they are saving health care providers over $250,000 per day. Another use-case [48], namely, General Electric, utilizes Apache Apex[70] for ingesting and analysing machine data from thousands of disparate sources in order to achieve the ingestion of data in real time, monitoring all of the IoT devices with sub-millisecond latency and zero data loss. Besides real-time processing, the case study of MapR [49], a digital media analytics company need to apply batch processing techniques for business intelligence to analyse and report behaviour of their customers weekly and monthly.

Data size: One of the most important factors when considering a big-data tool is the size of the data used in processing [50]. The preference may be for tools that support in-memory processing, such as Apache Spark, to avoid disk read/writes in order to increase the overall speed of the big-data application. One may also prefer to use on-disk processing, such as the Hadoop Distributed File System (HDFS), which is relatively slower than in-memory processing. However, it is necessary to avoid storing data on a disk in some cases in order to reduce latency and data regulations. For example, in the solution-brief of Indian railways [51], they have some performance issues on-disk processing to handle more than 3-millions of users in their e-ticket system. They determined that adding new servers or upgrading their hardware capabilities would not solve their performance issues. Therefore, they designed a completely new solution based on in-memory processing. They are currently able to handle 200.000 concurrent users without impacting the performance whilst, the old system would crash more than 40.000 concurrent users. Another use case use-case [52] of Barclays, which is one of the notable banks in the United Kingdom. In their case, they need to reduce the latency of big-data jobs from hours to seconds. They built a scalable, in-memory and reactive architecture to explore data and develop high-quality implementations. According to their need, they utilized Alluxio, which is an in-memory distributed file system rather than preferring HDFS on-disk storage. They are currently reaching the raw data immediately at every iteration, and they have reduced the overall waiting time of query results from hours to seconds.

Platform independency: The platform independency or interoperability of a big-data tool should also be considered when choosing proper big-data tools when there already exists a big-data architecture and it is necessary to integrate another tool to this architecture to

[70] Apache Apex, https://apex.apache.org/ (accessed 27 September 2017).

improve the data-analytics capability because there is no one-size-fits-all solution[34]. The data analytics or data-storage tools should also work compatibly with the distributed processing tool in the architecture. For example, an online machine-learning tool, SAMOA,[71] can easily be integrated into an application that uses stream-processing tools such as Apache Storm or Samza; however, it cannot be utilized in a batch-processing tool such as Spark. Tools such as H20 or MLlib, which is the native machine-learning library of Spark, should be preferred as a machine-learning tool for batch-processing architectures. As an example use-case [53], Samsung decided to move Apache Mesos technology to support development on their SAMI project for connected devices. The SAMI was a progressing project seeking a highly scalable platform-as-a-service solution. With the help of the platform independency of Mesos, they were able to integrate their project into this resource-management tool. Another use-case is about Apache Beam,[72] which is a unified programming model to define batch and streaming data-processing pipelines that are portable across a diverse set of runtime platforms such as Apex, Spark and Google DataFlow.[73] The big data and cloud-integration software provider Talend [54] proposed a data-preparation architecture with Apache Beam to bring a portable approach. They preferred Apache Beam because it mitigates the need to rewrite applications as new innovations are introduced and integration styles need to be alternated.

Data-storage model: Big data is a collection of large, complex, unstructured and continuous data from a large number of usually disparate data sources, and it is difficult to process this data using traditional database management tools or conventional data-processing approaches [55,56]. For example, a case-study of Cision [57] which is a cloud-based public-relations company, had built their software based on a SQL database management system, however, the utilized SQL system had reached its limits through massive amount of media data. They mentioned that, the SQL database had become more difficult to manage, backing it up, storing it and running reports. Therefore, they moved to open-source NoSQL solutions to have scalable and flexible database management system. In today's world, structured data constitutes only 5 per cent of the existing data [32,58]. The big-data applications generally bring together a diverse set of data sources to extract valuable knowledge, and these data sources may generate data in different data-storage models, such as graph-based data from social networks, key-value-based data from mobile applications, document-based data from web applications or time-series-based data from IoT devices. To this end, the software architects need to be aware of the data type that is ingested by the big-data application to decide the storage tool that is best suited to that data type. For example, Apache Accumulo[74] is a column-oriented database, Titan[75] is a graph-oriented database, CouchDB[76] is a document-oriented database and OpenTSDB[77] is a time-series database. For example, Yahoo utilizes [59] a key-value store database, HBase, for Yahoo Mail and Yahoo search to deliver

[71] Apache Samoa, https://samoa.incubator.apache.org/ (accessed 27 September 2017).
[72] Apache Beam, https://beam.apache.org/ (accessed 27 September 2017).
[73] Google Cloud DataFlow, https://cloud.google.com/dataflow/ (accessed 27 September 2017).
[74] Apache Accumulo, https://accumulo.apache.org/ ((accessed 27 September 2017).
[75] Titan, http://titan.thinkaurelius.com/ (accessed 27 September 2017).
[76] Apache CouchDB, http://couchdb.apache.org/ (accessed 27 September 2017).
[77] OpenTSDB, http://opentsdb.net/ (accessed 27 September 2017).

real-time performance. On the other hand, Facebook prefers [60] Apache Giraph,[78] a graph-oriented database, to build a model of Facebook users with connections between them that can represent almost anything.

In this section, we have discussed big-data tools and which of these tools can be utilized to process high-volume, fast-moving and diverse data sets. Moreover, an overall solution in the form of reference architecture for big-data analytics is depicted. However, beyond these technical issues, the main issue of big-data analytics extracting high-quality knowledge [61] to drive decision-making is a key requirement. Therefore, organizations should be aware of the quality of their data sources; because of the Garbage-In, Garbage-Out (GIGO) principle, poor quality of data will result in poor quality of output and will be a waste of valuable assets, time and money. Moreover, the culture of the company, and the skills of the developers and end-users need to be taken into account. As a result, companies can improve their productivity and performance by leveraging state-of-the-art big data technologies to exploit vast amount of data, but first, they have to change their decision-making culture [62].

4. **Discussion**

4.1. Problems of architecture development in big data

As we conducted our literature review and systematic review for the tools, we identified the major problems that an organization typically faces when building its own big-data analytics architecture. First, after filtering out the most important tools, we reviewed 241 open-source tools in Apache and GitHub that can be utilized as part of the big-data architecture of an organization. There is an abundance of tools available; however, there is no single best tool for a particular component in the architecture. As Google's 2015 article [63] points out, the tools have both strengths and shortcomings. An organization should choose from among the alternatives the most appropriate tool, considering not only the characteristics of the data to be analysed, but also its business strategy and operation domain. Google's article also points out that none of the shortcomings of the tools is intractable, and a tool's shortcoming may be diminished over time as the tool's maturity increases. The maturity of a tool is a risk for an organization since businesses incur costs when they try to change their source codes to run on newer versions. Not choosing the right tool for a particular task is another risk that needs to be avoided. The latest research focuses on tools providing abstractions on multiple data-processing platforms. For instance, programs built with Apache Beam can utilize Apache Spark and Apache Flink[79] as runners. Apache SAMOA provides machine-learning libraries to be used with Apache Storm and Apache Samza. Piglet [64] is also worth mentioning. It translates commands written with Pig Latin to be run on Apache Spark, Apache Flink and Apache Storm.

Choosing the most suitable tool for a particular job is important, but the technical challenge is not limited to this. There are also domain-specific challenges where the lower-level

[78] Apache Giraph, http://giraph.apache.org/ (accessed 27 September 2017).
[79] Apache Flink, http://flink.apache.org/ (accessed 5 October 2017).

programming model complexities and deployment problems make these tools suitable for programmers who have experience and knowledge in data science. On the other hand, there are people who have expertise and deep knowledge in a specific domain, and these people don't know how to utilize these tools. This technical barrier hinders the adoption of big-data tools as part of business processes [65]. To bridge the gap between domain-specific knowledge and data science, data-flow-based visual programming models are emerging. The streams framework [66,67] introduces an abstraction layer for domain experts to design and define processes without writing any code. Domain experts use two-dimensional surfaces (e.g. tablets) to sketch out processes in an interactive way. According to this concept, data is processed as it flows between existing building-blocks and this defines a streaming application. The process execution layer is independent from the process design layer. Depending on the process to be executed, the programmer can select from a set of execution platforms, which include Apache S4 and Apache Storm.

Apart from technical and domain-specific challenges, there are also firm-specific soft challenges when developing an architecture for big data in a business. Information produced from big-data analysis is of little use if managers lack the ability to foresee the value of results. As opposed to technical skills, managerial skills are deep-rooted in an organization. Developing a data-driven organizational culture is another challenge where members of the organization from all tiers incorporate insights from big-data analytics into their decision-making activities. This may be the hardest challenge, since people rely on their past experience or their superiors rather than data when making critical decisions [55]. Considering a firm that provides big-data analytics services, customers, who are the established firms, may not be able to perceive the value of big data and may question the value of big data if they don't understand the differentiation that the organization creates by utilizing big data in its business processes.

4.2. Research implications

This study is an attempt to delineate, classify and explain actively developed open-source tools that address the components of the big-data analytics life cycle, while keeping in mind the managerial perspective. We systematically reviewed the available open-source tools for big data and put them together in a taxonomy. The system for the review process revealed a method that can be used to track changes in this domain. The taxonomy revealed a simple and understandable architecture for big-data analytics. This section discusses the implications of this study from the perspective of various stakeholders of big-data.

This study originates on the systematic review of open-source tools. Academia can see the state-of-the-art tools, the gaps in research, and tools that are mature enough to be used as part of research. For technical personnel, it will help to determine the tool to be used for a particular implementation. In addition to the systematic review, we included case studies so that executives, as well as mid-level managers and operational staff, can see how some of these tools are utilized as part of the business processes of other firms. This represents the first step in establishing a data-driven culture in an organization.

We introduced a method to track changes in open-source big-data solutions, which is important since big data is a very active domain of research, and in the next couple of years we expect major changes. The tools in this article provide a snapshot of today, but in a couple of years academia will be able to use the method utilized in this article to obtain the latest snapshot before commencing research.

The proposed open-source big-data analytics architecture provides a comprehensive picture of the big-data analytics life cycle. An established firm trying to develop a strategy can use this reference architecture to come up with its own big-data analytics architecture according to its organizational requirements. As opposed to technical studies in the big-data domain, which attempts to develop architecture, we have tried to keep the architecture as simple as possible. Specifically, managers who lack the skills to foresee the value of big data in a business strategy can obtain a basic understanding of the concepts in big data. Commercial big-data solution providers can see the capability they lack and focus on that capability or collaborate with smaller firms to provide a similar solution. Accordingly, there are opportunities for small and medium-sized enterprises, who can provide services to established firms using some of the tools introduced, addressing the gaps in a larger architecture.

4.3. Limitations

This study has certain limitations. First, we reviewed only the open-source tools and excluded commercial solutions. The authors of the study have the best hands-on experience with some popular open-source big-data analytics tools, and we believe that the available open-source tools should be the critical components of a comprehensive architecture in an organization. In addition, the list of open-source tools is available for search in Apache and GitHub, with a variety of licences to develop a robust system for the review. This is not the case for commercial tools. Finally, most of the commercial tools provide a framework to address not a particular component in the proposed architecture but multiple components or the complete big-data analytics architecture of an organization.

Second, we endeavoured to include as many case studies as possible to explain how some of these tools are used in production. Also, we were unable to include a case study for all of these tools in this study. We have placed all of the tools in our taxonomy and the table can be found in the Appendix.

Finally, this is a managerial article as much as a technical one. It does not focus on the technical characteristics of the individual tools, such as ease of deployment, data-processing speed, windowing semantics, fault tolerance, correctness, message-delivery semantics, and so on. Furthermore, we have not investigated the interoperability of the tools when putting them together in the architecture. Nonetheless, the contributions of this study are still valuable to technical personnel, since it provides a comprehensive snapshot of the tools and the method used to obtain this snapshot in future when building an implementation.

5. **Conclusion**

Despite the abundance of open-source tools available in the big-data domain, newer tools never cease to emerge, and we do not expect this phenomenon to change in the near future. Studying the challenges of developing an organizational big-data architecture with the existing tools, we can foresee where the industry will focus its research efforts in this domain. This being the case, organizations should still try to build their own big-data architecture and exploit the potential of the available open-source tools instead of utilizing well-known and imposed commercial big-data tools. Overcoming the technical challenges in doing so is rewarding. Commercial big-data solution providers more or less rely on the same set of limited open-source tools that may or may not fit the nature of the analytics tasks in the business. Furthermore, no one can capture the domain-specific knowledge better than the organization itself. Developing the architecture would also help better decision-making, as the process would build a data-driven culture and develop the right managerial skills.

This study provides a comprehensive snapshot of the available open-source tools in a reference data-analytics architecture. While portraying the technical aspects, the paper also considers the managerial perspective by introducing cases as much as possible. To truly take advantage of insights from big-data analytics, an organization should focus on building the technical capabilities, as well as bridging the gap between the technical capabilities and firm-specific softer resources. These resources include the right set of managerial skills, a data-driven culture and domain-specific knowledge. Discounting the technical personnel and academia, the target audience of this study is expected to be all tiers of managers and operatives within an organization.

References

[1] K. Xie, Y. Wu, J. Xiao, Q. Hu, Value co-creation between firms and customers: The role of big data-based cooperative assets, Inf. Manag. 53 (2016) 1034–1048. doi:10.1016/j.im.2016.06.003.

[2] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, Machine Learning : The High-Interest Credit Card of Technical Debt, NIPS 2014 Work. Softw. Eng. Mach. Learn. (2014) 1–9. doi:10.1007/s13398-014-0173-7.2.

[3] T.H. Davenport, J. Dyché, Big Data in Big Companies, (2013). http://resources.idgenterprise.com/original/AST-0109216_Big_Data_in_Big_Companies.pdf (accessed August 18, 2017).

[4] S. Landset, T.M. Khoshgoftaar, A.N. Richter, T. Hasanin, A survey of open source tools for machine learning with big data in the Hadoop ecosystem, J. Big Data. 2 (2015) 24. doi:10.1186/s40537-015-0032-1.

[5] A. Oussous, F.Z. Benjelloun, A. Ait Lahcen, S. Belfkih, Big Data technologies: A survey, J. King Saud Univ. - Comput. Inf. Sci. (2017). doi:10.1016/j.jksuci.2017.06.001.

[6] C.-H. Chen, C.-L. Hsu, K.-Y. Tsai, Survey on Open Source Frameworks for Big Data Analytics, in: Third Int. Conf. Electron. Softw. Sci., 2017: p. 74.

[7] G.C. Fox, J. Qiu, S. Kamburugamuve, S. Jha, A. Luckow, HPC-ABDS high performance computing enhanced apache big data stack, in: Proc. - 2015 IEEE/ACM 15th Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2015, IEEE, 2015: pp. 1057–1066.

doi:10.1109/CCGrid.2015.122.
[8] P. Grover, A.K. Kar, Big Data Analytics: A Review on Theoretical Contributions and Tools Used in Literature, Glob. J. Flex. Syst. Manag. 18 (2017) 1–27. doi:10.1007/s40171-017-0159-3.
[9] P. Mikalef, I.O. Pappas, J. Krogstie, M. Giannakos, Big data analytics capabilities: a systematic literature review and research agenda, Inf. Syst. E-Bus. Manag. (2017) 1–32.
[10] U. Sivarajah, M.M. Kamal, Z. Irani, V. Weerakkody, Critical analysis of Big Data challenges and analytical methods, J. Bus. Res. 70 (2017) 263–286. doi:10.1016/j.jbusres.2016.08.001.
[11] D. Singh, C.K. Reddy, A survey on platforms for big data analytics, J. Big Data. 2 (2015) 8. doi:10.1186/s40537-014-0008-6.
[12] Apache Hadoop, (n.d.). http://hadoop.apache.org/ (accessed September 26, 2017).
[13] AMPLab – UC Berkeley, (n.d.). https://amplab.cs.berkeley.edu/software/ (accessed September 26, 2017).
[14] J. Qiu, S. Jha, A. Luckow, G.C. Fox, Towards HPC-ABDS: An Initial High-Performance Big Data Stack, ACM 1. 22 (2014). doi:10.1145/0000000.0000000.
[15] W. Inoubli, S. Aridhi, H. Mezni, A. Jung, An Experimental Survey on Big Data Frameworks, (2016). https://arxiv.org/pdf/1610.09962.pdf (accessed July 24, 2017).
[16] S. Nadal, V. Herrero, O. Romero, A. Abelló, X. Franch, S. Vansummeren, D. Valerio, A software reference architecture for semantic-aware Big Data systems, Inf. Softw. Technol. 90 (2017) 75–92. doi:10.1016/j.infsof.2017.06.001.
[17] P. Pääkkönen, D. Pakkala, Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems, Big Data Res. 2 (2015) 166–186. doi:10.1016/j.bdr.2015.01.001.
[18] B. Geerdink, A Reference Architecture for Big Data Solutions, 8th Int. Conf. Internet Technol. Secur. Trans. (2013) 71–76. doi:10.1109/ICITST.2013.6750165.
[19] J. Klein, R. Buglak, D. Blockow, T. Wuttke, B. Cooper, K. Town, A reference architecture for big data systems in the national security domain, in: Proc. 2nd Int. Work. BIG Data Softw. Eng. - BIGDSE '16, 2016: pp. 51–57. doi:10.1145/2896825.2896834.
[20] D. Tranfield, D. Denyer, P. Smart, Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review, Br. J. Manag. 14 (2003) 207–222. doi:10.1111/1467-8551.00375.
[21] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele Univ. 33 (2004) 28. doi:10.1.1.122.3308.
[22] Guidelines for performing Systematic Literature Reviews in Software Engineering, (2007). https://pdfs.semanticscholar.org/e62d/bbbbe70cabcde3335765009e94ed2b9883d5.pdf (accessed September 27, 2017).
[23] GitHub Octoverse 2016, (2016). https://octoverse.github.com/ (accessed September 27, 2017).
[24] M.A. Akyol, M.O. Gökalp, K. Kayabay, P.E. Eren, A. Koçyiğit, A Context Aware Notification Architecture Based on Distributed Focused Crawling in the Big Data Era, in: Springer, Cham, 2017: pp. 29–39. doi:10.1007/978-3-319-65930-5_3.
[25] Open Sourcing Twitter Heron, (2016).

https://blog.twitter.com/engineering/en_us/topics/open-source/2016/open-sourcing-twitter-heron.html (accessed October 4, 2017).
[26] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J.M. Patel, K. Ramasamy, S. Taneja, Twitter Heron, in: Proc. 2015 ACM SIGMOD Int. Conf. Manag. Data - SIGMOD '15, ACM Press, New York, New York, USA, 2015: pp. 239–250. doi:10.1145/2723372.2742788.
[27] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber, Bigtable, ACM Trans. Comput. Syst. 26 (2008) 1–26. doi:10.1145/1365815.1365816.
[28] J. Dean, S. Ghemawat, Simplified data processing on large clusters, Sixth Symp. Oper. Syst. Des. Implement. 51 (2004) 107–113. doi:10.1145/1327452.1327492.
[29] T. Akidu, A. Balikov, K. Bekiroglu, S. Chernyak, J. Haberman, R. Lax, S. McVeety, D. Mills, P. Nordstrom, S. Whittle, T. Akidau, K. Bekiroğlu, MillWheel: Fault-Tolerant Stream Processing at Internet Scale, Proc. VLDB Endow. 6 (2013) 734–746. doi:10.14778/2536222.2536229.
[30] G. Malewicz, M.H. Austern, A.J.. Bik, J.C. Dehnert, I. Horn, N. Leiser, G. Czajkowski, Pregel, in: Proc. 2010 Int. Conf. Manag. Data - SIGMOD '10, ACM Press, New York, New York, USA, 2010: p. 135. doi:10.1145/1807167.1807184.
[31] C. Chambers, A. Raniwala, F. Perry, S. Adams, R.R. Henry, R. Bradshaw, N. Weizenbaum, FlumeJava, in: PLDI, ACM Press, New York, New York, USA, 2010: pp. 363–375. doi:10.1145/1806596.1806638.
[32] A. Gandomi, M. Haider, Beyond the hype: Big data concepts, methods, and analytics, Int. J. Inf. Manage. 35 (2015) 137–144. doi:10.1016/j.ijinfomgt.2014.10.007.
[33] A. Labrinidis, H. V. Jagadish, Challenges and opportunities with big data, Proc. VLDB Endow. 5 (2012) 2032–2033. doi:10.14778/2367502.2367572.
[34] H. Hu, Y. Wen, T.S. Chua, X. Li, Toward scalable systems for big data analytics: A technology tutorial, IEEE Access. 2 (2014) 652–687. doi:10.1109/ACCESS.2014.2332453.
[35] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop Distributed File System, in: 2010 IEEE 26th Symp. Mass Storage Syst. Technol., IEEE, 2010: pp. 1–10. doi:10.1109/MSST.2010.5496972.
[36] H. Li, A. Ghodsi, M. Zaharia, E. Baldeschwieler, S. Shenker, I. Stoica, Tachyon: Memory Throughput I/O for Cluster Computing Frameworks, (2013). http://people.eecs.berkeley.edu/~haoyuan/papers/2013_ladis_tachyon.pdf (accessed September 27, 2017).
[37] H. Li, A. Ghodsi, M. Zaharia, S. Shenker, I. Stoica, Tachyon, in: Proc. ACM Symp. Cloud Comput. - SOCC '14, 2014: pp. 1–15. doi:10.1145/2670979.2670985.
[38] B. Hindman, A. Konwinski, A. Platform, F.-G. Resource, M. Zaharia, Mesos: A platform for fine-grained resource sharing in the data center, Proc. …. (2011) 32. doi:10.1109/TIM.2009.2038002.
[39] M. Zaharia, M.J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, I. Stoica, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, Apache Spark: a unified engine for big data processing, Commun. ACM. 59 (2016) 56–65. doi:10.1145/2934664.
[40] A. Toshniwal, J. Donham, N. Bhagat, S. Mittal, D. Ryaboy, S. Taneja, A. Shukla, K.

Ramasamy, J.M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, Storm@twitter, Proc. 2014 ACM SIGMOD Int. Conf. Manag. Data - SIGMOD '14. (2014) 147–156. doi:10.1145/2588555.2595641.
[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, (2016). http://arxiv.org/abs/1603.04467 (accessed September 27, 2017).
[42] J. Kreps, N. Narkhede, J. Rao, Kafka: a Distributed Messaging System for Log Processing, ACM SIGMOD Work. Netw. Meets Databases. (2011).
[43] R. Khare, D. Cutting, K. Sitaker, A. Rifkin, Nutch : A Flexible and Scalable Open-Source Web Search Engine, 14th Int. Conf. World Wide Web (WWW 2005). (2005) 1–10. doi:10.1101/gad.11.7.926.
[44] F.P. Junqueira, I. Kelly, B. Reed, Durability with BookKeeper, ACM SIGOPS Oper. Syst. Rev. 47 (2013) 9–15. doi:10.1145/2433140.2433144.
[45] A.R. Hevner, S.T. March, J. Park, S. Ram, Design Science in Information Systems Research, MIS Q. 28 (2004) 75–105. doi:10.2307/25148625.
[46] R. Divate, S. Sah, M. Singh, High Performance Computing and Big Data, in: Springer, Cham, 2018: pp. 125–147. doi:10.1007/978-3-319-53817-4_6.
[47] The Healthcare Challenge, (2017). http://www.pentaho.com/sites/default/files/uploads/resources/016-072_cs_healthcare.v6.pdf (accessed October 13, 2017).
[48] Apache Apex enables GE to transform industrial internet of things (IIoT) data into opportunity, (2017). https://www.datatorrent.com/wp-content/uploads/2017/01/GE-Case-Study-1.pdf (accessed September 27, 2017).
[49] MapR, comScore Reliability Processes Over 1.7 Trillion Internet & Mobile Events Every Month on MapR, 2014. https://mapr.com/resources/comscore-reliably-processes-over-17-trillion-internet-and-mobile-events-every-month-mapr/ (accessed October 13, 2017).
[50] C.-W. Tsai, C.-F. Lai, H.-C. Chao, A. V. Vasilakos, Big data analytics: a survey, J. Big Data. 2 (2015) 21. doi:10.1186/s40537-015-0030-3.
[51] Indian Railways: Distributed In-Memory Data Management Solution Improves the Capacity and Availability of New E-Ticketing System, (2017). https://content.pivotal.io/case-studies/indian-railways (accessed October 13, 2017).
[52] Alluxio: Accelerate Spark Jobs from Hours to Seconds, (2016). https://www.alluxio.com/resources/making-the-impossible-possible-with-alluxio-accelerate-spark-jobs-from-hours-to-seconds (accessed September 27, 2017).
[53] Samsung is powering the Internet of Things with Mesos and Marathon - Mesosphere, (2017). https://mesosphere.com/blog/samsung-is-powering-the-internet-of-things-with-mesos-and-marathon/ (accessed September 27, 2017).
[54] Talend Introduces the First Apache Beam Powered Big Data Preparation Solution - Talend Real-Time Open Source Data Integration Software, (2017). https://www.talend.com/about-us/press-releases/talend-introduces-first-apache-beam-

powered-big-data-preparation-solution/ (accessed September 27, 2017).
[55] M. Gupta, J.F. George, Toward the development of a big data analytics capability, Inf. Manag. 53 (2016) 1049–1064. doi:10.1016/j.im.2016.07.004.
[56] V. Mayer-Schönberger, K. Cukier, Big data: A revolution that will transform how we live, work, and think, Houghton Mifflin Harcourt, 2013.
[57] MapR, Cision Scales Their Business and Opens New Opportunities with MapR Solution | MapR, (2017). https://mapr.com/resources/cision-scales-their-business-and-opens-new-opportunities-mapr-solution/ (accessed October 13, 2017).
[58] Data, data everywhere | The Economist, (2010). http://www.economist.com/node/15557443 (accessed October 7, 2017).
[59] Yahoo! HBase, (2017). http://yahoohadoop.tumblr.com/post/161742444781/hbase-goes-fast-and-lean-with-the-accordion (accessed October 5, 2017).
[60] Scaling Apache Giraph, (2013). https://www.facebook.com/notes/facebook-engineering/scaling-apache-giraph-to-a-trillion-edges/10151617006153920/ (accessed October 5, 2017).
[61] I. Spiegler, Technology and knowledge: Bridging a "generating" gap, Inf. Manag. 40 (2003) 533–539. doi:10.1016/S0378-7206(02)00069-1.
[62] A. McAfee, E. Brynjolfsson, T.H. Davenport, Big data: the management revolution, Harv. Bus. Rev. 90 (2012) 60–68.
[63] T. Akidau, E. Schmidt, S. Whittle, R. Bradshaw, C. Chambers, S. Chernyak, R.J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, The dataflow model, Proc. VLDB Endow. 8 (2015) 1792–1803. doi:10.14778/2824032.2824076.
[64] S. Hagedorn, K.-U. Sattler, Piglet – Interactive and Platform Transparent Analytics for RDF & Dynamic Data, Proc. 25th Int. Conf. Companion World Wide Web. (2016) 187–190. doi:10.1145/2872518.2890530.
[65] M.O. Gokalp, K. Kayabay, M.A. Akyol, P.E. Eren, A. Kocyigit, Big Data for Industry 4.0: A Conceptual Framework, in: 2016 Int. Conf. Comput. Sci. Comput. Intell., IEEE, 2016: pp. 431–434. doi:10.1109/CSCI.2016.0088.
[66] C. Bockermann, H. Blom, The streams Framework (Report), Tech. Univ. Dortmund. (2012) 1–65. http://www.jwall.org/streams/.
[67] C. Bockermann, A visual programming approach to big data analytics, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer International Publishing, 2014: pp. 393–404. doi:10.1007/978-3-319-07626-3_36.

APPENDIX – A

	Name	Source	Website	Architectural Component
1.	Aperture Tiles	GitHub	https://github.com/unchartedsoftware/aperture-tiles	Data analysis
2.	OpenRefine	GitHub	https://github.com/OpenRefine/OpenRefine	Data pre-processing
3.	Data Cleaner	GitHub	https://github.com/datacleaner/DataCleaner	Data pre-processing
4.	Talend Open Studio	GitHub	https://github.com/Talend/tbd-studio-se	Data pre-processing
5.	Genie-Netflix	GitHub	https://netflix.github.io/genie/	Cluster management
6.	Chronos	GitHub	https://github.com/mesos/chronos	Cluster management
7.	Wherehow	GitHub	https://github.com/linkedin/WhereHows/wiki	Data pre-processing
8.	PrestoDB	GitHub	https://github.com/prestodb/presto	Data analysis
9.	HiBench	GitHub	https://github.com/intel-hadoop/HiBench	Governance & security
10.	Click House	GitHub	https://github.com/yandex/ClickHouse	Storage
11.	StreamSets-Data collector	GitHub	https://github.com/streamsets/datacollector	Data pre-processing
12.	Lumify	GitHub	https://github.com/lumifyio/lumify	Visualization
13.	Simba	GitHub	https://github.com/InitialDLab/Simba	Data analysis
14.	IndexR	GitHub	https://github.com/shunfei/indexr	Storage
15.	Hydrograph	GitHub	https://github.com/capitalone/Hydrograph	Data ingestion
16.	Plywood	GitHub	https://github.com/implydata/plywood	Visualization
17.	Smart Storage Management	GitHub	https://github.com/Intel-bigdata/SSM	Storage
18.	SpringXD	GitHub	https://github.com/spring-projects/spring-xd	Governance & security
19.	CKAN	GitHub	https://github.com/ckan/ckan	Data pre-processing
20.	ElasticSearch	GitHub	https://github.com/elastic/elasticsearch	Application
21.	Geomesa	GitHub	https://github.com/locationtech/geomesa	Data analysis
22.	Pentaho	GitHub	https://github.com/pentaho/big-data-plugin	Distributed data processing & programming
23.	Thrill	GitHub	https://github.com/thrill/thrill	Distributed data processing & programming
24.	GridDB	GitHub	https://github.com/griddb/griddb_nosql	Storage
25.	HPCC	GitHub	https://github.com/hpcc-systems/HPCC-Platform	Distributed data processing & programming
26.	FlashX	GitHub	https://github.com/flashxio/FlashX	Data analysis
27.	MOA	GitHub	https://github.com/Waikato/moa	Data analysis
28.	JStorm	GitHub	https://github.com/alibaba/jstorm	Distributed data processing & programming
29.	Riemann	GitHub	https://github.com/riemann/riemann	Distributed data processing & programming
30.	Tigon	GitHub	https://github.com/caskdata/tigon	Distributed data processing & programming
31.	Riko	GitHub	https://github.com/nerevu/riko	Distributed data processing & programming
32.	BoomFilters	GitHub	https://github.com/tylertreat/BoomFilters	Data pre-processing

33.	SensorBee	GitHub	https://github.com/sensorbee/sensorbee	Distributed data processing & programming
34.	Automi	GitHub	https://github.com/vladimirvivien/automi	Distributed data processing & programming
35.	Kubernetes	GitHub	https://github.com/kubernetes/kubernetes	Cluster management
36.	Squall	GitHub	https://github.com/epfldata/squall	Data analysis
37.	Goka	GitHub	https://github.com/lovoo/goka	Distributed data processing & programming
38.	SpringCloudDataFlow	GitHub	https://github.com/spring-cloud/spring-cloud-dataflow	Distributed data processing & programming
39.	Tron	GitHub	https://github.com/Yelp/Tron	Cluster management
40.	KilrWeather	GitHub	https://github.com/killrweather/killrweather	Application
41.	RapidMiner	GitHub	https://github.com/rapidminer/rapidminer-studio	Data analysis
42.	Esper	GitHub	https://github.com/espertechinc/esper	Data analysis
43.	Drools	GitHub	https://github.com/kiegroup/drools	Data analysis
44.	GraphJET	GitHub	https://github.com/twitter/GraphJet	Distributed data processing & programming
45.	Refarch	GitHub	https://github.com/awslabs/lambda-refarch-fileprocessing	Application
46.	Mondrian	GitHub	https://github.com/pentaho/mondrian	Data analysis
47.	Godot	GitHub	https://github.com/nodejitsu/godot	Data analysis
48.	PigPen	GitHub	https://github.com/Netflix/PigPen	Distributed data processing & programming
49.	Kibana	GitHub	https://github.com/elastic/kibana	Visualization
50.	Disco	GitHub	https://github.com/discoproject/disco	Distributed data processing & programming
51.	Infovore	GitHub	https://github.com/paulhoule/infovore	Distributed data processing & programming
52.	Redisson	GitHub	https://github.com/redisson/redisson	Governance & security
53.	Gleam	GitHub	https://github.com/chrislusf/gleam	Distributed data processing & programming
54.	Glow	GitHub	https://github.com/chrislusf/glow	Distributed data processing & programming
55.	NSQ	GitHub	https://github.com/nsqio/nsq	Data ingestion
56.	Metamorphosis	GitHub	https://github.com/killme2008/Metamorphosis	Data ingestion
57.	Jafka	GitHub	https://github.com/adyliu/jafka	Data ingestion
58.	Disque	GitHub	https://github.com/antirez/disque	Data ingestion
59.	Akka	GitHub	https://github.com/akka/akka	Governance & security
60.	Open Messaging	GitHub	https://github.com/openmessaging/openmessaging	Data ingestion
61.	VerneMQ	GitHub	https://github.com/erlio/vernemq	Data ingestion
62.	Cherami-Server-Client	GitHub	https://github.com/uber/cherami-server	Data ingestion
63.	Machinery	GitHub	https://github.com/RichardKnop/machinery	Data ingestion
64.	Vitess	GitHub	https://github.com/youtube/vitess	Cluster management
65.	Airpal	GitHub	https://github.com/airbnb/airpal	Visualization

66.	Cascalog	GitHub	https://github.com/nathanmarz/cascalog	Data analysis
67.	Cascading	GitHub	https://github.com/Cascading/cascading	Data analysis
68.	Parkour	GitHub	https://github.com/damballa/parkour	Distributed data processing & programming
69.	Druid	GitHub	https://github.com/druid-io/druid/	Storage
70.	Onyx	GitHub	https://github.com/onyx-platform/onyx	Distributed data processing & programming
71.	Scalding	GitHub	https://github.com/twitter/scalding	Data analysis
72.	SummingBird	GitHub	https://github.com/twitter/summingbird	Distributed data processing & programming
73.	Ceph	GitHub	https://github.com/ceph/ceph	Distributed file system
74.	Baidu File System	GitHub	https://github.com/baidu/bfs	Distributed file system
75.	SeaweedFS	GitHub	https://github.com/chrislusf/seaweedfs	Distributed file system
76.	GlusterFS	GitHub	https://github.com/gluster/glusterfs	Distributed file system
77.	QFS	GitHub	https://github.com/quantcast/qfs	Distributed file system
78.	XtreemFS	GitHub	https://github.com/xtreemfs/xtreemfs	Distributed file system
79.	Hyperdrive	GitHub	https://github.com/mafintosh/hyperdrive	Distributed file system
80.	Ambry	GitHub	https://github.com/linkedin/ambry	Distributed file system
81.	LizardFS	GitHub	https://github.com/lizardfs/lizardfs	Distributed file system
82.	FastDFS	GitHub	https://github.com/happyfish100/fastdfs	Distributed file system
83.	Dat-Node	GitHub	https://github.com/datproject/dat-node	Application
84.	MooseFS	GitHub	https://github.com/moosefs/moosefs	Distributed file system
85.	Azkaban	GitHub	https://github.com/azkaban/azkaban	Cluster management
86.	Schedoscope	GitHub	https://github.com/ottogroup/schedoscope	Cluster management
87.	Luigi	GitHub	https://github.com/spotify/luigi	Governance & security
88.	Serf	GitHub	https://github.com/hashicorp/serf	Governance & security
89.	Fineagle	GitHub	https://github.com/twitter/finagle	Governance & security
90.	Tensorflow	GitHub	https://github.com/tensorflow/tensorflow	Data analysis
91.	MLPack	GitHub	https://github.com/mlpack/mlpack	Data analysis
92.	Conjecture	GitHub	https://github.com/etsy/Conjecture	Data analysis
93.	Photon-ML	GitHub	https://github.com/linkedin/photon-ml	Data analysis
94.	DMLC	GitHub	https://github.com/dmlc/dmlc-core	Data analysis
95.	H20	GitHub	https://github.com/h2oai/h2o-3	Data analysis
96.	DSSTNE	GitHub	https://github.com/amzn/amazon-dsstne	Data analysis
97.	Angel	GitHub	https://github.com/Tencent/angel	Data analysis
98.	Oryx	GitHub	https://github.com/OryxProject/oryx	Data analysis
99.	Fregata	GitHub	https://github.com/TalkingData/Fregata	Data analysis
100.	Zen	GitHub	https://github.com/cloudml/zen	Data analysis
101.	BenchML	GitHub	https://github.com/szilard/benchm-ml	Data analysis
102.	Stream Alert	GitHub	https://github.com/airbnb/streamalert	Supporting tools

103.	Fenzo	GitHub	https://github.com/Netflix/Fenzo	Cluster management
104.	Redis	GitHub	https://github.com/antirez/redis	Storage
105.	Alluxio	GitHub	https://github.com/Alluxio/alluxio	Distributed file system
106.	TIDB	GitHub	https://github.com/pingcap/tidb	Storage
107.	Titan	GitHub	https://github.com/thinkaurelius/titan	Storage
108.	OpenTSDB	GitHub	https://github.com/OpenTSDB/opentsdb	Storage
109.	TIDB	GitHub	https://github.com/pingcap/tikv	Storage
110.	Crate	GitHub	https://github.com/crate/crate	Storage
111.	RQLite	GitHub	https://github.com/rqlite/rqlite	Storage
112.	ActorDB	GitHub	https://github.com/biokoda/actordb	Storage
113.	JanusGraph	GitHub	https://github.com/JanusGraph/janusgraph	Storage
114.	AtlasDB	GitHub	https://github.com/palantir/atlasdb	Storage
115.	CurioDB	GitHub	https://github.com/stephenmcd/curiodb	Storage
116.	Ceres	GitHub	https://github.com/graphite-project/ceres	Storage
117.	Hydra	GitHub	https://github.com/addthis/hydra	Distributed data processing & programming
118.	RethinkDB	GitHub	https://github.com/rethinkdb/rethinkdb	Storage
119.	Tera	GitHub	https://github.com/baidu/tera	Storage
120.	Scylla	GitHub	https://github.com/scylladb/scylla	Storage
121.	DGraph	GitHub	https://github.com/dgraph-io/dgraph	Storage
122.	Bolt	GitHub	https://github.com/boltdb/bolt	Storage
123.	BuntDB	GitHub	https://github.com/tidwall/buntdb	Storage
124.	Voldemort	GitHub	https://github.com/voldemort/voldemort/tree/master	Storage
125.	SummitDB	GitHub	https://github.com/tidwall/summitdb	Storage
126.	Mist	GitHub	https://github.com/Hydrospheredata/mist	Governance & security
127.	Secor	GitHub	https://github.com/pinterest/secor	Governance & security
128.	Jubatus	GitHub	https://github.com/jubatus/jubatus	Data analysis
129.	PipelineDB	GitHub	https://github.com/pipelinedb/pipelinedb	Data analysis
130.	StreamCQL	GitHub	https://github.com/HuaweiBigData/StreamCQL	Data analysis
131.	Redash	GitHub	https://github.com/getredash/redash	Application
132.	Bokeh	GitHub	https://github.com/bokeh/bokeh	Visualization
133.	Rakam-IO	GitHub	https://github.com/rakam-io/rakam	Application
134.	Countly	GitHub	https://github.com/Countly/countly-server	Application
135.	Finagle	GitHub	https://github.com/twitter/finagle	Supporting tools
136.	Elephant-Bird	GitHub	https://github.com/twitter/elephant-bird	Governance & security
137.	Kapacitor	GitHub	https://github.com/influxdata/kapacitor	Application
138.	Streaming Benchmark	GitHub	https://github.com/yahoo/streaming-benchmarks	Governance & security
139.	Riak	GitHub	https://github.com/basho/riak_kv	Storage

140.	Hstore	GitHub	https://github.com/apavlo/h-store/tree/release-2016-06	Storage
141.	Suro	GitHub	https://github.com/Netflix/suro	Data ingestion
142.	LogStash	GitHub	https://github.com/elastic/logstash	Data ingestion
143.	ElephantDB	GitHub	https://github.com/nathanmarz/elephantdb	Storage
144.	Apache Accumulo	Apache website	https://accumulo.apache.org	Storage
145.	Apache Airavata	Apache website	http://airavata.apache.org/	Cluster management
146.	Apache Ambari	Apache website	http://ambari.apache.org	Governance & security
147.	Apache Apex	Apache website	http://apex.apache.org/	Distributed data processing & programming
148.	Apache AsterixDB	Apache website	http://asterixdb.apache.org/	Data pre-processing
149.	Apache Atlas	Apache website	http://atlas.apache.org/	Governance & security
150.	Apache Avro	Apache website	http://avro.apache.org/	Data pre-processing
151.	Apache Bahir	Apache website	http://bahir.apache.org/	Supporting tools
152.	Apache Beam	Apache website	https://beam.apache.org/	Distributed data processing & programming
153.	Apache Bigtop	Apache website	http://bigtop.apache.org/	Governance & security
154.	Apache BookKeeper	Apache website	http://bookkeeper.apache.org/	Governance & security
155.	Apache Calcite	Apache website	https://calcite.apache.org/	Data analysis
156.	Apache CarbonData	Apache website	http://carbondata.apache.org/	Data pre-processing
157.	Apache Cassandra	Apache website	http://cassandra.apache.org	Storage
158.	Apache Chukwa	Apache website	http://chukwa.apache.org/	Data ingestion
159.	Apache CloudStack	Apache website	http://cloudstack.apache.org	Cluster management
160.	Apache CouchDB	Apache website	http://couchdb.apache.org/	Storage
161.	Apache Crunch	Apache website	http://crunch.apache.org/	Supporting tools
162.	Apache Curator	Apache website	http://curator.apache.org/	Governance & security
163.	Apache DataFu	Apache website	http://datafu.incubator.apache.org/	Distributed data processing & programming
164.	Apache Drill	Apache website	http://drill.apache.org/	Data analysis
165.	Apache Eagle	Apache website	http://eagle.apache.org/	Governance & security
166.	Apache Edgent	Apache website	http://edgent.incubator.apache.org/	Supporting tools
167.	Apache Falcon	Apache website	http://falcon.apache.org/	Distributed data processing & programming
168.	Apache Flink	Apache website	http://flink.apache.org/	Distributed data processing & programming

169.	Apache Fluo	Apache website	http://fluo.apache.org	Supporting tools
170.	Apache Flume	Apache website	http://flume.apache.org/	Data ingestion
171.	Apache Gearpump	Apache website	https://gearpump.apache.org/overview.html	Distributed data processing & programming
172.	Apache Geode	Apache website	http://geode.apache.org	Governance & security
173.	Apache Giraph	Apache website	http://giraph.apache.org/	Distributed data processing & programming
174.	Apache Gobblin	Apache website	http://gobblin.incubator.apache.org/	Data ingestion
175.	Apache Gora	Apache website	http://gora.apache.org	Storage
176.	Apache Griffin	Apache website	http://griffin.incubator.apache.org	Data pre-processing
177.	Apache Hadoop	Apache website	http://hadoop.apache.org/	Distributed data processing & programming
178.	Apache Hama	Apache website	http://hama.apache.org/	Distributed data processing & programming
179.	Apache HAWQ	Apache website	http://hawq.incubator.apache.org/	Data analysis
180.	Apache HBase	Apache website	http://hbase.apache.org	Storage
181.	Apache Helix	Apache website	http://helix.apache.org/	Cluster management
182.	Apache Heron	Apache website	https://twitter.github.io/heron	Distributed data processing & programming
183.	Apache Horn	Apache website	http://horn.incubator.apache.org/	Data analysis
184.	Apache Hive	Apache website	http://hive.apache.org/	Data analysis
185.	Apache Hivemall	Apache website	http://hivemall.incubator.apache.org/	Data analysis
186.	Apache HTrace	Apache website	http://htrace.incubator.apache.org/	Governance & security
187.	Apache Ignite	Apache website	http://ignite.apache.org/	Distributed data processing & programming
188.	Apache Impala	Apache website	http://impala.incubator.apache.org/	Data analysis
189.	Apache Kafka	Apache website	http://kafka.apache.org/	Data ingestion
190.	Apache Kerby	Apache website	http://directory.apache.org/kerby/	Governance & security
191.	Apache Knox	Apache website	http://knox.apache.org/	Supporting tools
192.	Apache Kudu	Apache website	http://kudu.apache.org/	Data analysis
193.	Apache Kylin	Apache website	http://kylin.apache.org/	Data analysis
194.	Apache Lens	Apache website	http://lens.apache.org/	Data analysis
195.	Apache MADLib	Apache website	http://madlib.apache.org	Data analysis
196.	Apache Mahout	Apache website	http://mahout.apache.org/	Data analysis

197.	Apache Mesos	Apache website	http://mesos.apache.org/	Cluster management
198.	Apache MetaModel	Apache website	http://metamodel.apache.org/	Data analysis
199.	Apache Milagro	Apache website	http://milagro.incubator.apache.org/	Governance & security
200.	Apache Metron	Apache website	http://metron.apache.org/	Governance & security
201.	Apache MRQL	Apache website	http://mrql.incubator.apache.org/	Data analysis
202.	Apache Myriad	Apache website	http://myriad.incubator.apache.org/	Cluster management
203.	Apache Nifi	Apache website	http://nifi.apache.org	Data ingestion
204.	Apache Nutch	Apache website	http://nutch.apache.org/	Application
205.	Apache Omid	Apache website	http://omid.incubator.apache.org	Supporting tools
206.	Apache Oozie	Apache website	http://oozie.apache.org/	Cluster management
207.	Apache OODT	Apache website	http://oodt.apache.org	Governance & security
208.	Apache OpenWhisk	Apache website	http://openwhisk.incubator.apache.org/	Supporting tools
209.	Apache ORC	Apache website	https://orc.apache.org/	Storage
210.	Apache Parquet	Apache website	http://parquet.apache.org/	Storage
211.	Apache Phoenix	Apache website	http://phoenix.apache.org/	Data analysis
212.	Apache Pig	Apache website	http://pig.apache.org/	Data analysis
213.	Apache Pulsar	Apache website	http://pulsar.incubator.apache.org	Data ingestion
214.	Apache Ranger	Apache website	http://ranger.apache.org/	Governance & security
215.	Apache REEF	Apache website	http://reef.apache.org/	Cluster management
216.	Apache River	Apache website	http://river.apache.org/	Supporting tools
217.	Apache RocketMQ	Apache website	http://rocketmq.incubator.apache.org/	Data ingestion
218.	Apache Rya	Apache website	http://rya.incubator.apache.org/	Storage
219.	Apache S2Graph	Apache website	http://s2graph.incubator.apache.org/	Storage
220.	Apache SAMOA	Apache website	http://samoa.incubator.apache.org/	Data analysis
221.	Apache Samza	Apache website	http://samza.apache.org/	Distributed data processing & programming
222.	Apache Sentry	Apache website	http://sentry.apache.org/	Governance & security
223.	Apache SINGA	Apache website	http://singa.incubator.apache.org/	Data analysis
224.	Apache Slider	Apache website	http://slider.incubator.apache.org/	Cluster management

225.	Apache Spark	Apache website	http://spark.apache.org/	Distributed data processing & programming
226.	Apache Spot	Apache website	http://spot.incubator.apache.org/	Governance & security
227.	Apache Sqoop	Apache website	http://sqoop.apache.org/	Data ingestion
228.	Apache Storm	Apache website	http://storm.apache.org/	Distributed data processing & programming
229.	Apache SystemML	Apache website	http://systemml.apache.org/	Data analysis
230.	Apache Tajo	Apache website	http://tajo.apache.org/	Data analysis
231.	Apache Tephra	Apache website	http://tephra.incubator.apache.org/	Supporting tools
232.	Apache Tez	Apache website	http://tez.apache.org/	Cluster management
233.	Apache Thrift	Apache website	http://thrift.apache.org	Governance & security
234.	Apache Trafodion	Apache website	http://trafodion.incubator.apache.org/	Data analysis
235.	Apache Twill	Apache website	http://twill.apache.org/	Cluster management
236.	Apache VXQuery	Apache website	http://vxquery.apache.org/	Data analysis
237.	Apache Zeppelin	Apache website	http://zeppelin.apache.org/	Visualization
238.	Apache ZooKeeper	Apache website	http://zookeeper.apache.org/	Governance & security
239.	Apache Aurora	Apache website	http://aurora.apache.org	Cluster management
240.	Apache Solr	Apache website	http://lucene.apache.org/solr/	Application
241.	Apache Lucene	Apache website	https://lucene.apache.org/core/	Application