DESIGN PRINCIPLES FOR HADOOP-BASED PLATFORMS: A REVERSE-ENGINEERED DESIGN-SCIENCE APPROACH

Ravi S. Sharma, Zayed University
Stephen C. Wingreen, University of Canterbury
Satheesh B.T. Janarthanan, University of Canterbury

ABSTRACT

Big data has revolutionised industry in many fields and has emerged as an integral component of the 4th industrial revolution. Many organisations adopt big data for making significant strategic decisions. However, due to the availability of multiple big data platforms and analytical tools, organisations are faced with the challenge of developing inter-operable platforms. The objective of this research is to identify design principles and rules that may be effectively used in heterogeneous distributions of Hadoop-based big data platforms for both development and operations (DevOps). The methodology adopted is a "reverse-engineered design science research" approach for extracting the design principles and rules from artefacts. Three big data platforms were evaluated using this approach in order derive ten design principles and associated rules that aid in the implementation of a big data platform with emphasis on inter-operability. This is the theoretical contribution of the research. A practical contribution is the general guidelines for a reverse-engineered, design science research approach that supports the derivation and validation of effective implementation rules.

Keywords: Big Data Analytics, Hadoop Platform, Design Science Research, Reverse Engineering

INTRODUCTION

Big data has revolutionised cloud computing, data analytics and the Internet-of-Things and continues to make contributions to systems in healthcare, transportation, governance and education, to name a few (cf. Chen & Zhang, 2014). After an extensive review of scholarly work in the area, Hashem, et al. (2015) conclude that "addressing big data is a challenging and time-demanding task that requires a large computational infrastructure to ensure successful data processing and analysis" but "certain important aspects of storing and processing big data in cloud computing are yet to be solved." One such issue is addressing the interoperability of big data systems. This paper serves to address the issue of interoperability with the reverse engineering of design principles and rules so that practitioners may apply them in heterogeneous, distributed clound envinronments.

Big Data is sometimes referred to as the "new oil" for industries' success and transformation because of its ability to create business value for organisational growth (Sun, Cegielski, Jia & Hall, 2018), and its promise to make radical changes in aiding and optimising organisational decision-making processes, customer management, and business growth. However, according to Forbes, only 53% of organisations have adopted big data analytics (Columbus, 2017). A report by the Gartner Group (2018) identifies three limitations which prevent organisations from adopting big data and hinder its development to the fullest potential: strategy for data and analytics, thorough understanding the process of extracting the value from projects, and managing governance and risks related to big data. Therefore, companies need a strong grasp on their data and the related technologies before developing system requirements and selecting the correct analytics solution, if the greatest value from big data is to be achieved.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Big data primarily targets unstructured and semi-structured data, such as browsing history, logs, chats, messages etc., as opposed to traditional structured data (e.g. RDBMS) (Hashem, et al., 2015). Big data's role in industry is underscored by the expectation that unstructured data will grow to 80% of existing data by 2020 (D'Cruz, 2016). As data is growing exponentially the key challenge is to store and organise data for analysis. Big data analysis may provide insight into an organisation's customers if the users of big data understand the nuances and traps associated with the use of big data. The Hadoop platform has become one of the primary solutions for unstructured data, and there are many Hadoop variants being offered by vendors. Some data management vendors are now creating custom solutions based on local business requirements as innovation, service-oriented offerings (Erraissi, Belangour, & Tragha, 2017).

While the development of Big Data and Hadoop platforms has received increasing scholarly attention (cf Chen & Zhang, 2014; Hashem et al., 2015; Erraissi, Belangour & Tragha, 2017), considerably less research has investigated the design principles and design rules used in developing Hadoop solutions, partly because big data and analytics are a relatively recent development in industry, and practice research inherently lags behind industrial innovation. This corresponds to Path 1 in Figure 1 of (Kuechler & Vaishnavi, 2012), where the justificatory knowledge (theory) for developing the systems (big data solutions) is missing and such knowledge needs to be derived from the existing systems. We propose a "reverse-engineered design science research" (RE-DSR) approach to abductively extract design specifications of big data and analytics systems. We applied RE-DSR to three key big data analytics solutions identified from a set of Hadoop-based systems such as HortonWorks, Cloudera, MapR, IBM Infosphere BigInsights, Pivotal HD, Microsoft HD Insight (Erraissi, Belangour, & Tragha, 2017). The primary contribution of such RE-DSR artefacts are both to support organisations in the development of a heterogeneous big data platform and to assist researchers to keep pace with rapid innovations in general, and modern Hadoop big data systems in particular.

Following this introduction, the remainder of the paper is organised as follows. The background review section provides a detailed literature review and theoretical background about big data and big data platforms. The research methodology section discusses the RE-DSR methodology used for this research. The analysis and discussion section presents the results and significance of the findings. Finally, the conclusion section presents the research implications and the recommendations for future research.

BACKGROUND REVIEW

Big Data and Hadoop

Big data is critical to the development of intellectual assets in many industries, and therefore may assist in the next level innovation, productivity improvement, and competitive advantage (Chen & Zhang, 2015; (Saggi & Jain, 2018). Because of the rapid development of cloud computing, Internet of Things (IOT), and related sensing devices the generation of data has increased multi-fold and sparked interest across industry and business fields (Jin, W.Wah, Cheng, & Wang, 2015; Kiran, Murphy, Monga, Dugan, & Baveja). In the scholarly literature, the term big data is typically applied in the context of large companies like Yahoo, Google, and Facebook in discussions of the need to analyse large amounts of data collected by their systems. The "5V's" of big data are shown in Figure 1 from "Big Data Hadoop building blocks comparative study" (Erraissi, Belangour, & Tragha, 2017), which currently is the most commonly accepted framework, and includes the dimensions: volume, veracity, variety, velocity, and value.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

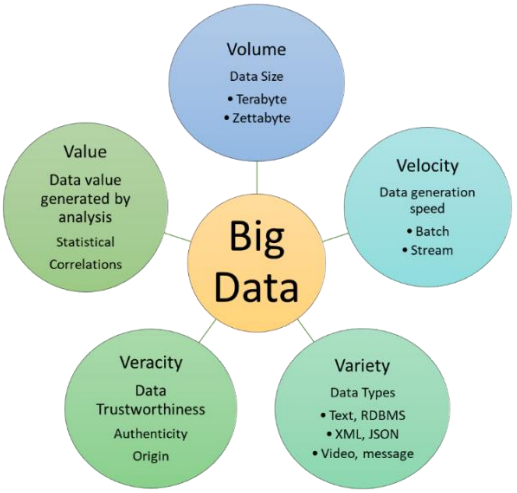

FIGURE 1
5 V'S OF BIG DATA PLATFORMS (ADOPTED FROM (SAGIROGLU & SINANC, 2013) (HASHEM, ET AL., 2015) (ISHWARAPPA & ANURADHA, 2015)

The data generated by modern systems is stored as structured data, unstructured data, and semi-structured data (Chunarkar-Patil & Bhosale, Big data analytics, 2018), though only 5% of the generated data is formatted as structured data, such as traditional RDBMS (Gandomi & Haider, 2015). The analysis of structured data is comparatively less cumbersome, often reported in the form of numbers, figures and transactions. Conversely unstructured data - such as images, text messages, user-generated data - requires specialised systems to analyse and process it. Semi-structured data, such as extensible markup language (XML), lies on the continuum between the two (Gandomi & Haider, 2015). Big data systems primarily target the unstructured and semi-structured data with a specialised set of tools and techniques.

Big Data Analytics

"Big data analytics" refers to the methods of mining a large set of data. Different types of technologies and tools are used, and there are multiple features and techniques involved which require thoughtful analysis before a system is implemented (Chunarkar-Patil & Bhosale, Big data analytics, 2018; Gandomi & Haider, 2015; Saggi & Jain, 2018). Because of this, selection of tools and techniques becomes a source of competitive advantage (Chunarkar-Patil & Bhosale, Big data analytics, 2018). It is therefore a critical success factor to understand both the type of data and the larger analytics landscape.

The process of extracting information from the big data can be done in five steps in two stages. The data management consists of: collecting, storing, and preparing the data for the analytics stage. The analytics stage consists of: analysing, and accruing the information available in the big data (Gandomi & Haider, 2015). The data management phase implements a specialised new storage type known as a "data lake", which is a database capable of storing all types of data in a raw format without transforming or processing any data. Data lakes provide an additional advantage of accessing data dynamically and in real time (Madera & Laurent, 2016). The analytics stage may be further segmented into three different categories: descriptive analysis, predictive analysis and prescriptive analysis (HU, WEN, CHUA, & LI, 2014), which are accomplished by a variety of analytical techniques, such as: structured data analytics, text analytics, network analytics, web analytics, multimedia analytics, mobile analytics, statistical analytics, artificial intelligence and complex SQL analysis (Gandomi & Haider, 2015; Chunarkar-Patil & Bhosale, Big data analytics, 2018; HU, WEN, CHUA, & LI, 2014). The basic requirements for analytical tools and systems are that they should be scalable and high speed (Chunarkar-Patil & Bhosale, Big data analytics, 2018). Large vendors are currently offering cloud solutions which can support big data systems, and tools for managing and

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

analysing vast quantities of data; for example, Netflix and Spotify (Heilig & Voß, 2017). However, capacity, cost and resources remain challenges (Kiran, Murphy, Monga, Dugan, & Baveja).

Hadoop Platforms

The Hadoop platform is one of the major solutions for managing big dataaccording to a study on Digging into Hadoop-based Big Data Architectures by (Erraissi, Belangour, & Tragha, 2017). Hadoop was invented by Doug Cutting and Mike Cafarella in 2005, as an open source system which was built largely using Java framework and inspired from the Google file system and Google's MapReduce framework. Hadoop is a cost-effective, high precision, scalable system capable of handling data in various formats such as structured tabular data, images, videos, files, audios, sensor data, and various others (Saraladevi, Pazhaniraja, Paul, Basha, & Dhavachelvan, 2015). Because of this, Hadoop has become the preferred storage and processing system for many data-heavy organisations (Alam & Ahmed, 2014). The Hadoop taxonomy is explained in Figure 2.

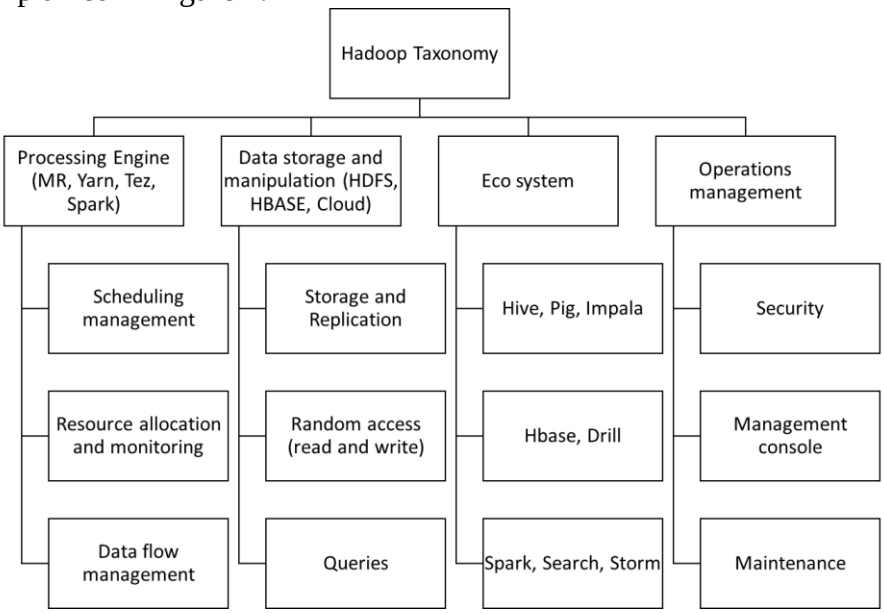

FIGURE 2
HADOOP TAXONOMY (ADOPTED FROM (POLATO, RÉ, GOLDMAN, & KON, 2014)

There are multiple Hadoop-based solutions currently available for organisations to select from as well as vendors who provide professional services. Examples of vendors include HortonWorks, Cloudera, MapR, IBM Infosphere BigInsights, Pivotal HD, Microsoft HD Insight, all of whom are developing new systems to improve their products as well as increase the usage of Hadoop solutions in various fields (Erraissi, Belangour, & Tragha, 2017)

Hadoop implements the Visualisation, Platform management, Storage, Security, Monitoring, Ingestion, and Data source layers to represent big data systems (Erraissi, Belangour, & Tragha, Meta-modeling of Data Sources and Ingestion Big Data layers, 2018), which we generalize to the following layers: Data ingestion, Data storage, Operations and management, Computing and processing, Data analytics and visualisation. As an example, in Figures 3, 4, and 5 we use a general framework to represent the component-based architecture of three current Hadoop-based distributions.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

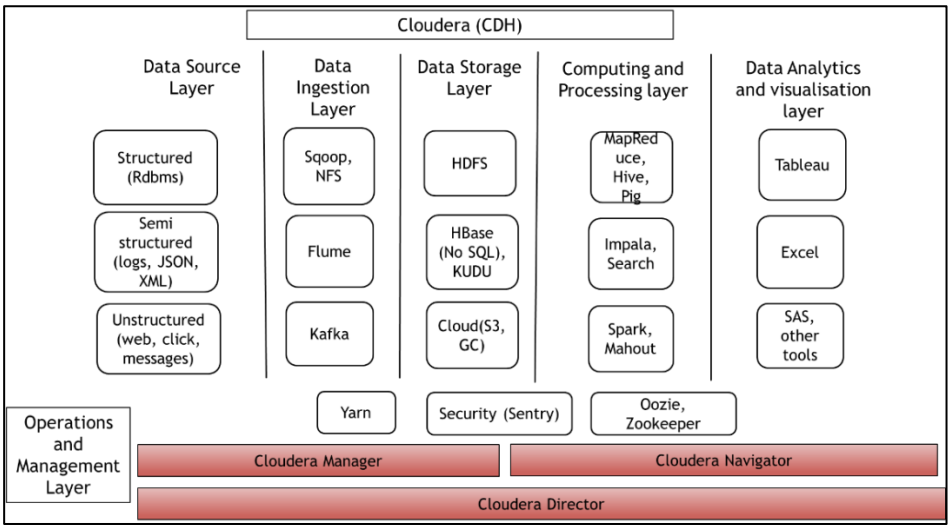

FIGURE 3
CLOUDERA DATA PLATFORM SOURCE: (CLOUDERA, N.D)

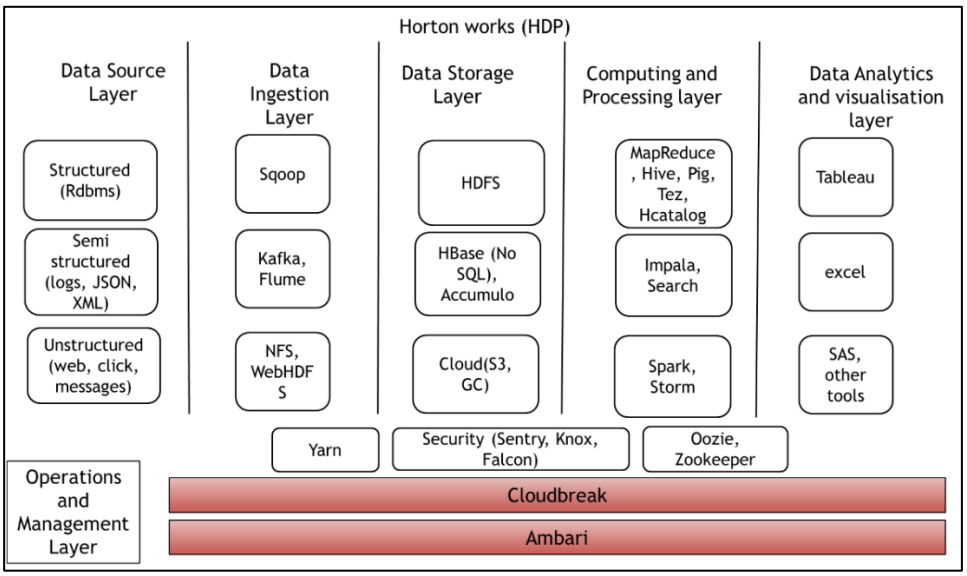

FIGURE 4
HORTONWORKS DATA PLATFORM SOURCE: (HORTONWORKS, N.D.)

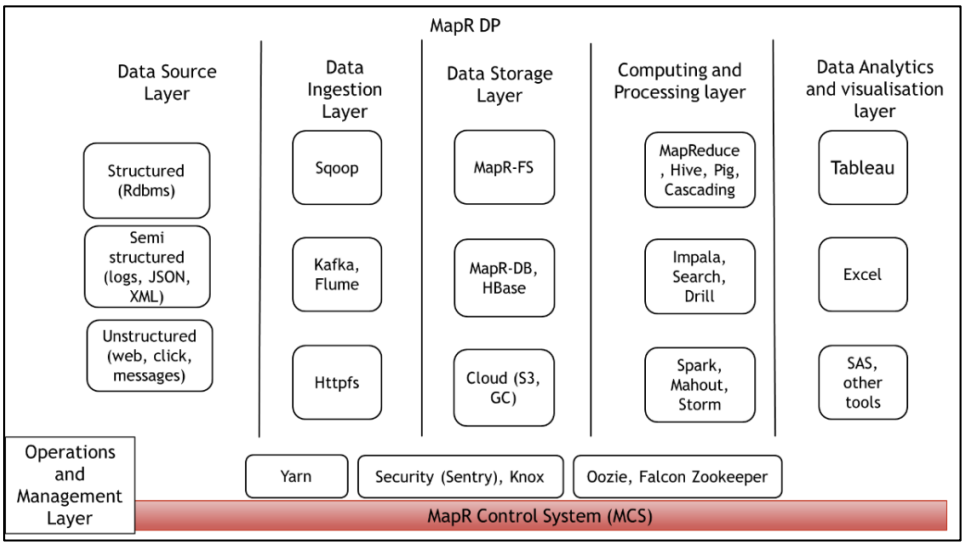

FIGURE 5
MAPR DATA PLATFORM SOURCE: (MAPR, N.D.)

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

The Role of Cloud-based Platforms

Some vendors offer cloud-based hosting as a solution for some deficiencies in the Hadoop platform, such as "low-level programming paradigm and schema, strictly batch processing, time skew and incremental computation" (Alam & Ahmed, 2014). AWS, Google, and Microsoft have also created their own proprietary big data platforms as a substitute for Hadoop systems (Veiga, Exposito, Pardo, Taboada, & Tourino, 2016), which include services such as: online data collection tools, database hosting, and a range of processing tools like MapReduce, Hive, and Spark. These vendors offer a virtual machine for hosting, computing, and managing the data. Virtual machines offer multiple advantages for the users such as elasticity, pay-as-you-go cost model, and multi-tenancy, all of which lessen the threats posed by volatile platform features (Kiran, Murphy, Monga, Dugan, & Baveja, 2015). The variety of possible cloud-based solutions poses a key challenge about which platform to choose. One of the objectives of this research will be to determine the design principles and rules that should be used to evaluate the various platforms.

RESEARCH METHODOLOGY

We define and use a special case of Design Science Research (DSR) by incorporating elements of reverse engineering to derive design principles and associated design rules from three big data platforms. Our objective in doing so was to elicit effective features for the implementation of Hadoop-based systems.

Design Science Research Methodology

We adopt the design science paradigm because it is consistent with our goals to develop artefact consisting of system design rules and principles which may be applied in practice or research (Hevner, March, Park, & Ram, 2004; Peffers, Tuunanen, Rothenberger & Chatterjee, 2007; Vaishnavi & Kuechler, 2015). A seminal paper on DSR proposes seven guidelines tabulated below.

Table 1
GUIDELINES FOR DESIGN SCIENCE RESEARCH (ADAPTED FROM (HEVNER, MARCH, PARK, & RAM, 2004))

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

In a later account of the subject, Gregor & Hevner (2013) state that most IT artefacts will have some level of abstraction which can be developed into a concrete material of usage; for example, an abstracted algorithm which may be developed into software. Sometimes an artefact will be a narrative in the form of design principles or technical rules which is the first level of

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

developing a better design theories or body of knowledge. It is important to maintain a certain level of abstraction in an artefact, since it helps to conceptualise various unstudied areas, and to generalise and validate an artefact as a solution to multiple problems. Specifically, in the case of our research, we will propose a general set of design rules and principles for Hadoop-based systems that may be applied to multiple big data problems.

Reverse Engineering

Reverse engineering is an approach used in the field of information technology to understand the design, architecture and processes of an established system, especially systems for which no documentation or limited documentation is available to the users. The reverse engineering process extracts information from the target system for multiple purposes, such as remodelling the system, solving key problems, or creating a better version of the system (Garg & Jindal, 2009).

More recently, reverse engineering has been used to examine successful software systems by creating high-level representations and models to explain the concepts or design principles of the system. This achieves many outcomes such as, satisfying customer needs, adopting the systems to the new business models, changes due to the legislation modification, protecting the system degradation and adopting with the systems to the new technological innovations (Brunelière, Cabot, Dupé, & Madiot, 2014). Reverse engineering is particularly useful as a method for understanding rapid innovations in emerging technologies about which very little may be known by anyone other than its designers, or for innovations in the early stages of growth where the requirements and best practices are still in development, as is the case in current big data and analytics systems.

Reverse Engineered - Design Science Research (RE-DSR) Methodology

Although the design science methodology is typically used to create new technological artefacts, we believe it may also be used effectively to reverse engineer existing technologies and their associated artefacts. The design science methodology provides effective guidelines and a framework for creating design artefacts (Hevner, March, Park, & Ram, 2004; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007; Vaishnavi & Kuechler, 2015), which may also be applied to the logic of reverse engineering. The application of DSR to reverse engineering, which we call RE-DSR, is intended to produce design artefacts that may be generalized to other domains of problems by information systems researchers and practitioners.

RE-DSR is a special case of DSR where design begins with a complete, validated artefact, and by abductive logic works backward toward the artefact's most likely design principles. Abductive logic reasons toward the most likely cause of an observation; according to Peirce (1997), abductive logic seeks an explanation that, if true, would make the observation in question a matter-of-course. In the case of RE-DSR the original design principles are an implicit theory for the eventually-developed and subsequently-observed system and its features, and the rigorous methods of DSR provide a set of analytical and developmental tools that may assist in the abductive logic of RE-DSR. Since the reverse engineering process is typically used to extract design principles and rules, it naturally lends itself to the use of abductive logic with DSR. We apply RE-DSR to the Hadoop big data platform. With the dual goals of extracting general design principles for Hadoop platforms, and developing a set of general guidelines for RE-DSR, Figure 6 explains the step-by-step processes followed in this research. We next discuss some general principles for RE-DSR in our description of Steps 1-6.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

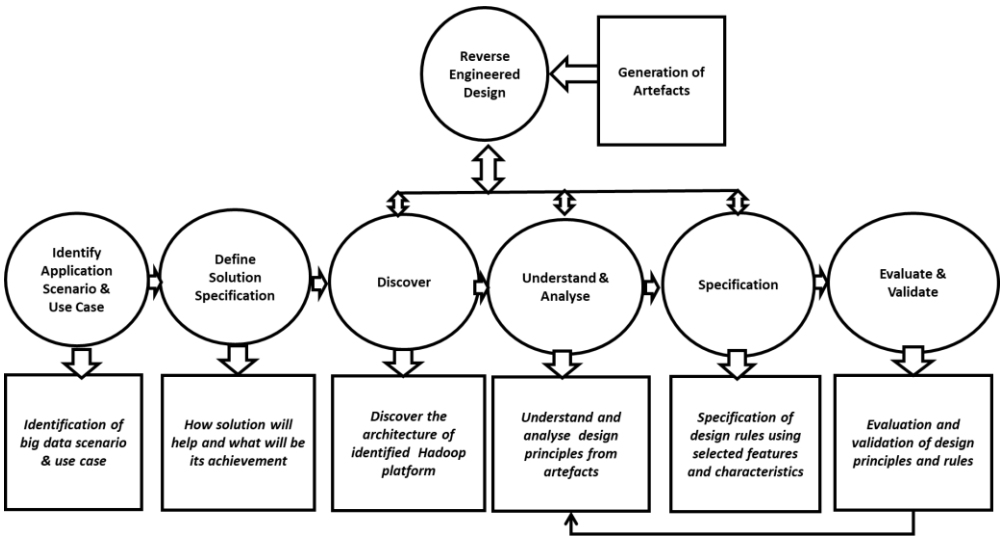

FIGURE 6
REVERSE-ENGINEERED DESIGN RESEARCH METHODOLOGY

Step 1 – Identify Application Scenario / Use Case

There are many big data solutions available in the market, and organisations have difficulty because they lack the necessary technical knowledge to select and implement them (Sang, Xu, & Vrieze). To further complicate matters, there are also multiple variants of the systems available in the market, many of which are still evolving (Erraissi, Belangour, & Tragha, 2017). To extract general principles and rules, it is desirable to begin RE-DSR by selecting the most established, mature, and functional systems available. In our research, we chose the Cloudera (Cloudera), Hortonworks (Hortonworks), and MapR (Mapr) Hadoop-based systems because of their reputation in industry.

Step 2 – Define Solution Specification

For any given artefact there should be a guiding set of design principles, but since no single artefact is a complete solution for its problem domain, it may therefore be necessary either to narrow the scope of the problem domain for RE-DSR or examine more than one artefact so as to achieve a more complete and generalizable set of design principles. The primary objective of this research is to generate an artefact consisting of the design principles and rules used in Hadoop data platforms as a guide for both academic researchers and practitioners engaged in the design of Hadoop-based big-data systems. We have chosen the three representative platforms mentioned in Step 1 so as to achieve a higher probability of deriving a more comprehensive and generalizable set of design principles.

Step 3 – Discover

A functional representation of the target system is used to discover its architecture. By an examination and analysis of the literature, published documentation, and direct experience with the systems, we identified a five-layered architecture for the Cloudera, Hortonworks and MapR systems we had selected for our investigation: data ingestion, data storage, data computation, data analysis and operations and management layer (Figure 7).

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

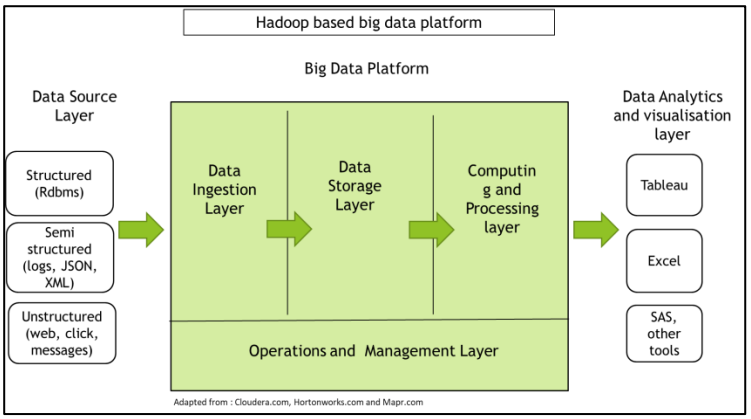

FIGURE 7
BIG DATA PLATFORM LAYERS (ADOPTED FROM ERRAISSI, BELANGOUR, & TRAGHA, (2018)

The examination and analysis also allowed us to identify the components involved in every layer of each platform (Figure 8).

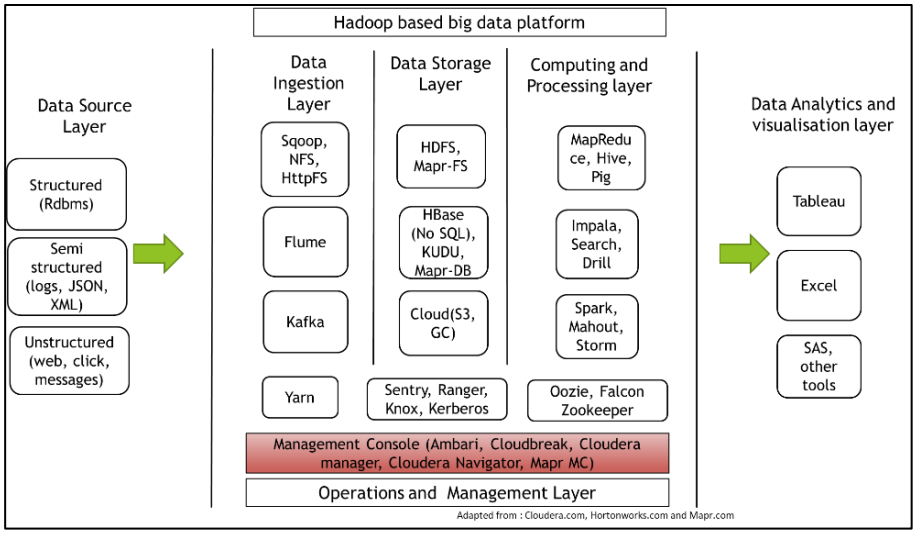

FIGURE 8
HADOOP REVERSE ENGINEERED WITH COMPONENTS (ADAPTED FROM (CLOUDERA, N.D.) (HORTONWORKS, N.D.) (MAPR, N.D.))

Step 4 – Understand & Analyse

Abductive logic is used to understand and analyze every component's input, output, and internal processing details. General principles are developed by reasoning, for example, "Component A's input, output, and processing details would be a matter-of-course if design principle B is true. Therefore we have reason to believe that principle B is true." Every component was analyzed in detail with the help of sequence and activity diagrams, and in this manner the first level of design principles were extracted.

Step 5 – Specification

Once the design principles were identified, the next step is to identify the associated design rules. Every principle will have its own associated rules. These rules were identified with the help of the use case as well as architecture diagrams.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Step 6 – Evaluate and Validate

The topic of evaluation and validation deserves some special attention in the context of RE-DSR. Since RE-DSR begins with a validated artefact, we may generally assume that design principles extracted from a validated artefact are themselves valid – assuming, of course, that we have correctly extracted the design principles. Vaishnavi & Kuechler (2015) devote an entire chapter to evaluation and validation. How do we know that we are correct in our RE-DSR? In abductive reasoning, there may be multiple theories that explain an observation, and the goal is to identify the most likely theory or theories. In the terminology of Vaishnavi & Kuechler (2015), abductive reasoning is listed in the DSR cycle as one of the cognitive reasoning techniques. But, if there are multiple related observations that all lead abductively to the same theoretical explanation, then we may have more confidence in the validity of the explanation. In the context of our research, we analysed three different Hadoop-based data platforms as a means to minimize the threat posed by reasoning from a single case. The identified principles and the rules thus derived will be explained in detail in the analysis and discussion section.

DESIGN PRINCIPLES AND DERIVED RULES FOR HADOOP PLATFORMS

In the reverse engineering process, we use abductive logic to deduce the design principles based on the components manifest in the design of the three Hadoop-based systems we considered (ie. Cloudera, MapR, and Hortonworks). In the first step of RE-DSR (identify application/ use case), we observed that both HDFS and MapReduce were so tightly coupled in the Hadoop architecture that the components were used by both applications (Hashem, et al., 2015). This architecture was suitable for batch processing alone, but with the growth of technology and data and the need for RDBMS/SQL-style data warehousing activities, real-time processing was not feasible until the development of new architectures such as YARN, Spark, and Tez (Vavilapalli, et al., 2013) that provided flexibility in processing (Gurusamy, Kannan, & Nandhini, 2017). For example, Apache Spark can replace MapReduce and work with HDFS; similarly, Yarn can work with both cloud as well as the HDFS system. Because of these inventions, organisations may choose components based on their input data source, and therefore our first design principle is derived from the flexibility of component choices.

Design Principle 1: Big Data Platforms Require Robust and Flexible Architecture or Framework

Based on their own unique business requirements, companies are using various operating systems such as Windows, Linux and rarely MacOS (iOS), and therefore a big data platform should support their existing operating environment. Hadoop-based big data platforms support both Windows and Linux operating system (Cloudera, n.d.) (Hortonworks, n.d.). Therefore, our first design rule is about operating system flexibility.

Accommodate Diverse Operating Systems Based on User Requirements

Big data platforms are divided into multiple layers based on functionalities such as the storage layer, ingestion layer, computing and processing layer and visualisation layer as depicted in Figure 7. There are multiple components available in each layer of the Hadoop platform, which may be integrated for the purpose of scalability, flexibility and interoperability. The Hadoop system is also widely available and has the highest market share, thus making it an ideal exemplar for a design rule, according to a "A Big Data Hadoop building blocks comparative study" (Mazumdar & Dhar, 2015; Erraissi, Belangour, & Tragha, 2017). Based on this feature our next design rule is that a flexible system should have integrated components.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Facilitate Reuse and Integration of the Different Big Data Architectural Layers

In the second step of RE-DSR (define solution specification), components of the indiviudal layers were reverse engineered and further principles and rules were extracted. Initially, organisations were analysing only the data collected from the online transaction processing system, which is referred to as structured data. With technological developments, many companies are now generating data in different formats such as semi-structured and unstructured. For example to understand the behaviour and browsing patterns of a customer, clickstream data may be used, or facial recognition may be used to understand the traffic, age and gender of a customer (Gandomi & Haider, 2015).

New processing engines such as Spark, Tez and Storm have been developed for analysing unstructured and semi-structured data (Chen & Zhang, 2014; Hashem, et al., 2015), which implies a requirement for ingesting all types of data into a big data platform. The Hadoop system supports all types of data ingestion as demonstrated in Figure 9.

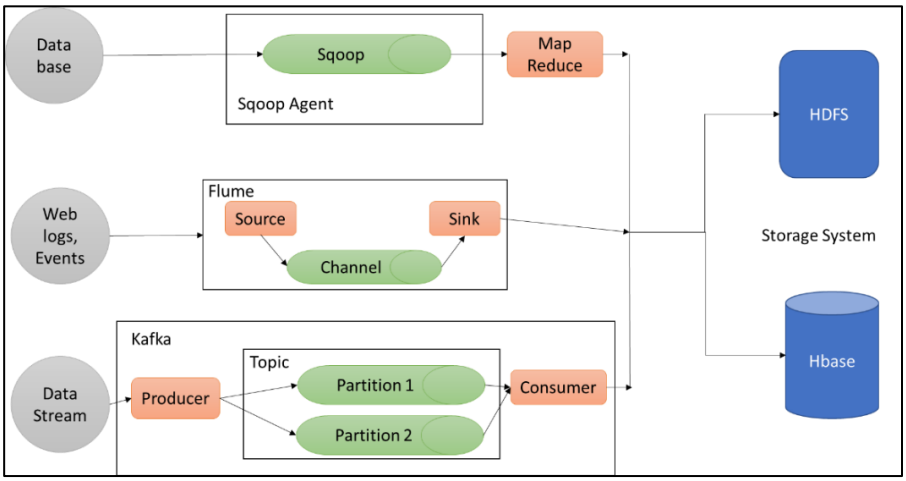

FIGURE 9
ARCHITECTURE OF HADOOP INGESTION COMPONENTS (ADOPTED FROM (CLOUDERA, N.D.) (HORTONWORKS, N.D.) (MAPR, N.D.))

Therefore, our next design principle is about data ingestion, specifically that a big data platform should be capable of ingesting diverse types of data.

Design Principle 2- The Ingestion Layer Should Handle Diverse Data Formats

Despite the surge of interest in unstructured and semi-structured data, relational databases still play a major role in the data analysis, and show no sign of becoming obsolete. Big data platforms, including Hadoop, support analytical tools such as Pig, Hive, Impala, and Sqoop for relational data. Therefore our next rule specifies that the ingestion layer should support the ingestion of relational data.

Accommodate Structured Data from Relational Databases in the Ingestion Layer

Initially, big data solutions were handling data in the batch mode, which means the data were not analysed in real time. Because of increasing interest in making real-time decisions, solutions for real-time data analysis were developed, such as Flume, Apache Spark, Apache Storm, etc. (Yaqoob, et al., 2016). Therefore our next design rule specifies that big data platforms should support both real-time and batch mode data ingestion.

Make Ingestion Layer Independent of Data Generation Modes

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Not all of the data collected may be useful for any given analysis (Hire, Gavande, Chovatiya, Tekale, & Patel, 2018). Organisations have a need to customize the system views for their various analytical requirements. Therefore, our next rule specifies that the ingestion layer should support custom selection of data types.

Support Custom User Data Type Configurations

One of the key components in big data solutions is the storage layer. Because of rapid growth in data generation capabilities, the storage component needs to be flexible and scalable in nature. Multiple storage platforms are available such as HDFS based on commodity hardware, cloud-based storage such as S3, Google cloud. Recently, cloud systems have increased in popularity on account of their flexibility, scalability and elasticity (Hashem, et al., 2015). Organisations may use either the commodity-based storage HDFS, or the latest cloud-based solutions such as Cloudera (6.0) (Cloudera, n.d.), Hortonworks (6.0) (Hortonworks, n.d.) and MapR (Mapr, n.d.), which make use of cloud systems such as AWS, Azure, and Google cloud. Because of this, our third design principle is about the independence of storage architecture.

Design Principle 3- Big Data Platform Should be Independent of Storage Architectures

Both Hadoop and cloud-based systems support two types of solutions depending on whether the storage is on-premises or externally hosted systems. Cloudera (Cloudera, n.d.), Hortonworks (Hortonworks, n.d.), MapR, AWS and Google cloud all support both the types of configurations, and therefore is the basis for our next design rule.

Accommodate Both On-Premises and Externally Hosted Storage

Generally, HDFS raw data is stored in text, JSON, CSV, and XML file formats. There are two drawbacks due to this: inefficient disk space usage and slower data processing. Additionally, Hadoop systems replicate data three times, which consumes even more storage space. New binary data formats such as Avro, parquet, ORC, RCFile may help in both the effective storage, reading, and processing of data (Figure 10) (PLASE, 2017). Therefore, the fourth design principle is about storage and data file formats.

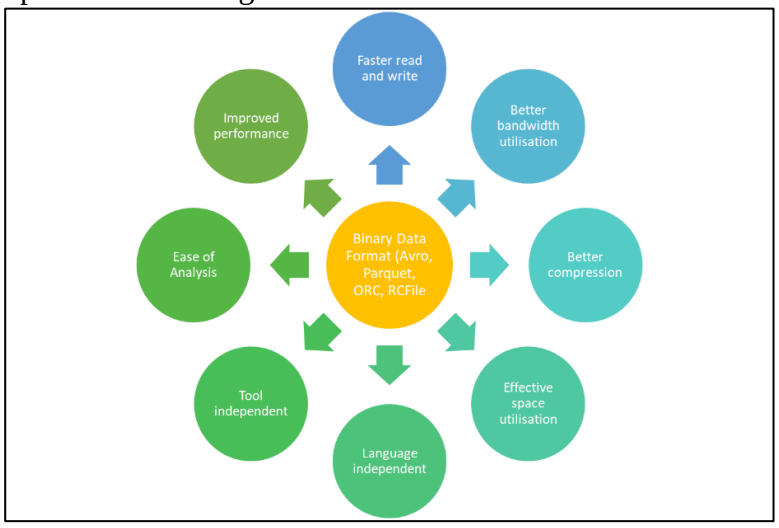

FIGURE 10
BENEFITS OF BINARY DATA FORMATS (ADAPTED FROM (PLASE, 2017) (WHITE, 2015))

Design Principle 4: Data Needs to be Stored in Open as well as Specific File Formats

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Any enterprise system should be able to access data in various file formats. HDFS stores data in different formats such as Avro, ORC, Parquet, RCFile (White, 2015) to optimize disk usage and processing by component systems such as Hive, Pig, Spark (PLASE, 2017). Therefore, the next design rule is about the inclusion of diverse file formats for analytical tools.

Make Computing, Processing and Analytics Components Process Multiple File Formats

Contingencies for hardware or system failure are a critical requirement for all big data platforms, including Hadoop systems, and therefore big data systems feature recovery for lost data (White, 2015, p. 5). Big data solutions offer many redundancy options to support data recovery. Hadoop systems replicate data a minimum of three times (White, 2015, p. 46), and AWS and Google replicate data in multiple cloud systems as requested by the originator. Therefore the fifth design principle of data redundancy is defined.

Design Principle 5: **Data Redundancy Should be Built-In to Ensure Platform Reliability**

Previous data analysis systems such as data warehousing, moved data from relational databases to their own analytical database where the analysis took place, which was a bandwidth-heavy and time-consuming activity. To overcome this disadvantage Hadoop was designed to bring the analysis to the data for processing (Chen & Zhang, 2014; White, 2015), which typically uses Yarn or MapReduce to select the data node, and then move the processing job to that node for execution. Therefore the sixth design principle of data locality is defined.

Design Principle 6: Data Locality Should be Built-In by Bringing the Analysis to Data

Another key requirement of the computation layer pertains to the automatic restart of failed jobs. In the Hadoop process, data is split into multiple blocks, with jobs being executed in parallel using distributed environments. Any one job failure could jeopardise the entire computation process. To avoid failure, basic rules are implemented to restart jobs in a different healthy node, thus ensuring that the computation is executed successfully. MapReduce implements the "shared nothing architecture" (White, 2015), which means there is no dependency between jobs, and therefore no risk of one failed job jeopardizing the entire process. Therefore, our next design rule addresses task resilience.

Provide Computation Resilience to Ensure Task Failure Prevention

Early big data analysis systems used a single process to analyse all of the data, which became less efficient as the data volume increased. Distributed and parallel processing were introduced by processing the analysis into multiple sub processes, each of which were then executed on a different machine with data specific to that sub-process; this significantly increased processing speed and efficiency (Humbetov, 2012).

Google was the first to use this model, and later MapReduce developed the same for their system (Humbetov, 2012), and it is now standard for many processing engines. Therefore, our seventh design principle is about distributed and parallel processing.

Design Principle 7: The Computing and Processing Layers Should Support Distributed and Parallel Processing

Big data platforms require data to be distributed and processed in parallel in multiple data nodes to expedite execution and make efficient utilisation of bandwidth. Hadoop systems

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

balance the workload by splitting the data into blocks of 64M or 128M and distributing the blocks to different data nodes for processing (Humbetov, 2012; Chen & Zhang, 2014). Therefore, our next design rule is about the use of parallel processing to manage the processing workload.

Store Data in Distributed Environments for Workload Balancing

Multiple departments in an organization often need to apply different analytical techniques to extract data for their own purpose, which implies that a big data platform should support the execution of parallel analytical techniques so as to avoid long wait times. Yarn was developed to address the inefficiencies in the scalability and flexibility of MapReduce, by supporting multiple tools like Pig, Hive, Storm, Spark that may execute different analytical tasks in parallel. Therefore, our next design rule is to reduce wait times by supporting parallel analytical techniques.

Allow Multiple Analytic Tools to Work in Parallel for Optimal Waiting Times

Sometimes the same data needs to be processed concurrently by multiple analytical tools. For example, different departments in an organization may need to work with the same data to generate a marketing report, a customer report, or for product development. Apache Falcon is a data life management tool that was developed to support the concurrent use of multiple analytical tools (Apache Falcon, n.d.). Because of this, our next design rule is about providing multiple users access to the same data concurrently.

Enable Multiple Analytic Tools to Concurrently Process the Same Data Sets

The need for greater and greater computing power also creates greater and greater latency, which may have negative effects on organizational performance, especially for time-sensitive activities (Zhang, Chen, Ooi, Tan, & Zhang, 2015; Chen & Zhang, 2014). In Hadoop systems, the primary cause of latency is the frequent reading and writing of data to the HDFS local storage, which stores data after each processing step for the next step in the process, and causes I/O process bottlenecking while the process waits for next processing thread (Zhang, Chen, Ooi, Tan, & Zhang, 2015).

These issues have been addressed by the development of in-memory computing, which reads the data only once from the HDFS storage and copies it to memory for the remainder of the computing process. Memory-based processing utilizes different types of memory to host a significant part of the database for execution or analysis (Chen & Zhang, 2014; Yaqoob, et al., 2016). Therefore, our eighth design principle of in-memory computation is defined.

Design Principle 8: Big Data Platform Should Support in Memory Computation

Big data platforms implement cache memory to support in-memory computing, first by increasing the speed of data access and a reduction of I/O processes, and secondly by storing intermediate results of computations in memory for subsequent computational jobs, rather than storing results in the HDFS disk or doing an entire re-computation of the data (Zhang, Chen, Ooi, Tan, & Zhang, 2015). New technologies such as Spark, Storm and Flink were developed to support in-memory computing. Spark's in-memory computation is shown in Figure 11 as example. The benefits of in-memory computation have been subsequently realised by machine learning. Therefore, our next design rule addresses in-memory computation with cache memory.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

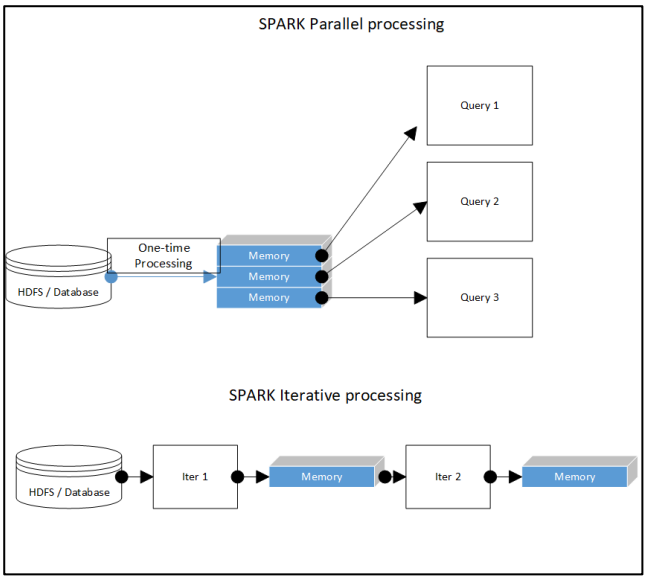

FIGURE 11
IN-MEMORY COMPUTATION – SPARK

Access Intermediate Data from Cache Memory for Further Analytics

Security is a critical concern for big data platforms because of how big data often represents many seemingly innocuous day-to-day activities that may reveal sensitive information when analyzed. For example, in a well-known case Target sent direct mail marketing materials about pregnancy products to a teenage girl because their analysis revealed that her shopping habits indicated she may be pregnant, and in fact she was, much to her father's surprise. Multiple levels of security are needed for big data platforms such as access control, authorisation control and encryption during data at rest, and data in motion or transit (R. PARMAR, ROY, BHATTACHARYYA, BANDYOPADHYAY, & KIM, 2017), as well as network and infrastructure, system auditing, and monitoring of events (Narayanan, 2013). Because of this, the threat of damages is immense (Mengke, Xiaoguang, Jianqiu, & Jianjian, 2016). Therefore, our ninth design principle addresses proper data security management.

Design Principle 9: Big Data Platform Should Include Data Protection Capabilities to Avoid Data Leakage and System Breaches

The Hadoop system uses multiple components for the security of its big data platforms. Sentry and Ranger are used for component authorisation level security, with an additional security component called Knox, which is used for firewall level security. Kerberos is another security component for user-level security. HDFS data encryption is used for data encryption to ensure data security (Cloudera, n.d.) (Hortonworks, n.d.) (Mapr, n.d.). Similarly, AWS uses 'AWS key management Services' and 'Amazon Macie' (AWS Amazon, n.d.) for security, and Google uses both 'Cloud security scanner' and 'Cloud platform security' (Cloud Google, n.d.) systems as security management components in their big data solutions. Therefore, our next design rule addresses platform security.

Provide Access, Authorisation, Authentication and Accounting Management of the System

Deploying and managing a large cluster of Hadoop-based big data platforms is cumbersome because of the user interface. A number of solutions have been developed to overcome this challenge. The Cloudera system, called Cloudera manager, supports the deployment and management of enterprise level Hadoop based platforms. Similarly, Apache

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Ambari supports cluster management and implements performance monitoring tools (Chung, et al., 2014). Cloudbreak may be used to implement Hadoop in cloud infrastructures. Therefore, our tenth design principle addresses the user interface for system management and implementation.

Design Principle 10: Superior UX for Performing Installation, Configuration and Management of Big Data Platform

Performance management is one of the key activities of Hadoop cluster management (Chung, et al., 2014). Ambari's Cloudera manager provides a rich set of tools to support operational activities and best practices (Hortonworks, n.d.). Therefore, our next key rule is about system performance management.

Provide User-Friendly Operations Management

Effective management of metadata involves multiple activities such as storing and extracting the metadata, separating the metadata, and data flow traffic. If the metadata is not managed effectively the throughput of data decreases as the number of concurrent users or system workload increases, even though the size of the metadata files may be small (Singh and Bawa, 2018). For this reason metadata storage is distributed, and several techniques such as sub-tree partitioning, consistent hashing, and hashing technique are implemented to manage it (Deshpande, 2017). Therefore, our next design rule addresses effective metadata management.

Make the system facilitate administration of metadata management such as metadata capture, storage, integration and governance

DISCUSSION AND CONCLUSION

Industry 4.0 or the 4th Industrial Revolution is witnessing the dawn of the big data era, and effective big data management is regarded as the key to success. Outperforming the competition with data-driven decision-making will be critical characteristics of emerging information systems and technologies. By reverse-engineering three variants of the Hadoop platform, this article identifies key design principles and rules that are characteristic of big data platforms. The identified design principles and rules are valuable to practitioners who wish to implement big data systems, and the RE-DSR method used to extract the principles and rules may be generally applied to other technological artefacts by design science researchers.

The use of RE-DSR to extract design principles and rules may be generalized to larger classes of technological problems beyond the Hadoop-based data platform problems investigated in this research. If we assume that an artefact is validated, then design principles and rules extracted from it must by definition also be valid, although this depends entirely on the correct use of abductive logic. Therefore abductive reasoning is central to RE-DSR since it is the logical engine by which an artefact's most likely design principles are derived. Building on the established practices of DSR (Vaishnavi & Kuechler, 2015), our research also provides a six-step framework for RE-DSR that may be used to guide future research.

Limited research has been done surrounding the design principles and rules of Hadoop-based Big Data platforms. This research also provides a framework that may be used to add structure to future research. As this research investigated three different big data platform for analysis, it may be relevant to future research about interoperability of big data systems, and may be able to support the creation of new innovations in big data platforms.

Our research may be refined by including the practical usage and operation of the big data platform and various types of data management tools. Practice-oreinted research may add more design-level information to the design rules and principles extracted by our research.

The RE-DSR study was mainly conducted on secondary data such as research articles,

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

third-party-users, and product-specific websites. There is hence some limitation in the evaluation and validation of the findings of this research. To provide greater degree of certainty and to show the utility and generality of the design principles and associated rules it would be fruitful to investigate whether other Hadoop platforms conform with the design principles and rules and to evaluate their weaknesses if they do not. Having extracted design principles and rules from one set of design artefacts of operational systems (ie Cloudera, Hortonworks, mapr) as training data for "modelling existing solutions", a "design experiment" with a test ensemble of artefacts of other Hadoop platforms might "predict" the efficacy of the ten design principles and their nested rules. Solutions architects knowledgeable about Hadoop platforms could evaluate the test set on the basis of compliance with the ten principles and rules using benchmarking.

Hence, we may conclude that the RE-DSR approach provided a means of generating principles and rules with which we may "desk check" Hadoop platforms in terms of the strengths and weaknesses of their functionalities. Deeper analysis will also reveal any inter-operability issues between legacy systems and new implementations. We claim that such guidelines will support architects and managers of heterogeneous Big Data environments in various DevOps tasks.

As we witness the transformation of business by Big Data, we are at the threshold of ubiquitous data platforms, including Hadoop-based solutions. The expectation is that such systems will continue to support industries such as internet of things, augmented reality, artificial intelligence to transform the cloud to the next level of data and analytics as services.

ACKNOWLEDGEMENT

The authors declare no conflict of interests. They further declare that their contributions to this article are in accordance with DORA. They gratefully acknowledge the constructive comments from Profs Vijay Vaishnavi (Georgia State U) and Al Hevner (U of South Florida).

REFERENCES

Ajila, S.A. (2002). A framework for reverse engineering process.

Alam, A., & Ahmed, J. (2014). Hadoop architecture and its issues. Paper presented at International Conference of *Computational Science and Computational Intelligence (CSCI)*, 288-291.

Bigdata, (2018). Retrieved from mattturck:

Brunelière, H., Cabot, J., Dupé, G., & Madiot, F. (2014). MoDisco: A model driven reverse engineering framework. *Information and Software Technology, 56*(8), 1012-1032.

Chunarkar-Patil, P., & Bhosale, A. (2018). Big data analytics. *Open Access J Sci. 2018, 2*(5):326–335.

Columbus, L. (2017). 53% of companies are adopting big data analytics.

D'Cruz, M. (2016). Navigating unstructured retail data storm.

Deshpande, P.P. (2017). *Hadoop Distributed File System: Metadata Management*.

Dhar, S., & Mazumdar, S. (2014). Challenges and best practices for enterprise adoption of big data technologies. *IEEE International Technology Management Conference (ITMC)*, 1-4.

Emmerik, M. (n.d.). Decompilation And Reverse Engineering.

Erraissi, A., Belangour, A., & Tragha, A. (2017). A big data hadoop building blocks comparative study. *International Journal of Computer Trends and Technology*, 36-40.

Erraissi, A., Belangour, A., & Tragha, A. (2017). Digging into hadoop-based big data architectures. *International Journal of Computer Science Issues (IJCSI), 14*(6), 52-59. at

Erraissi, A., Belangour, A., & Tragha, A. (2018). Meta-Modeling of data sources and ingestion big data layers.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*(2), 137-144.

Garg, M., & Jindal, M.K. (2009). Reverse engineering - roadmap to effective software design. *International Journal of Recent Trends in Engineering, 1*(2), 186.

Gartner. (2018). Gartner survey shows organizations are slow to advance in data and analytics.

Gregor, S., & Hevner, A.R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337-355.

Gurusamy, V., Kannan, S., & Nandhini, K. (2017). The real time big data processing framework: Advantages and limitations. *International Journal of Computer Science and Engineering (IJCSE), ISSN*, 2347-2693.

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems, 47*, 98-115.

Heilig, L., & Voß, S. (2017). Managing cloud-based big data platforms: A reference architecture and cost perspective. In *Big Data Management*, 29-45. Springer, Cham.

Hevner, A.R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information systems, 19*(2), 4.

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hire, M., Gavande, M., Chovatiya, K., Tekale, A., Patel, M.P., & Degadwala, S. (2018). *Study on big data challenges and opportunities*. d International Conference on Current Research Trends in Engineering and Technology, *4*(5), 424-429.

Hu, H., Wen, Y., Chua, T., & Li, X. (2014). Toward scalable systems for big data analytics: A technology tutorial. IEEE Access, *2*, 652-687.

Humbetov, S. (2012). Data-intensive computing with map-reduce and hadoop. Paper presented at the *6th International Conference*, IEEE, 1-5.

Jin, X., Wah, B.W., Cheng, X., & Wang, Y. (2015). Significance and challenges of big data research. *Big Data Research, 2*(2), 59-64.

Kiran, M., Murphy, P., Monga, I., Dugan, J., & Baveja, S.S. (2015). Lambda architecture for cost-effective batch and speed big data processing. Paper presented at the 2785-2792.

Kuechler, W., & Vaishnavi, V. (2012). "A framework for theory development in design science research: Multiple Perspectives." *Journal of the Association for Information Systems, 13*(6), 395-423.

Lorenz, K. (2008). Chapter 6. The Missing Link: Instructional Ethology, Methodology for Analyzing Learning Support in Games.

Madera, C., & Laurent, A. (2016). *The next information architecture evolution: The data lake wave*. In Proceedings of the 8th International Conference on Management of Digital EcoSystems, 174-180.

Mengke, Y.A.N.G., Xiaoguang, Z., Jianqiu, Z., & Jianjian, X. (2016). Challenges and solutions of information security issues in the age of big data. *China Communications, 13*(3), 193-202.

Narayanan, S. (2013). *Securing hadoop* (1st ed.). Birmingham: Packt Publishing.

Parmar, R.R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S.K., & Kim, T. (2017). Large-scale encryption in the hadoop environment: Challenges and solutions. *IEEE Access, 5*, 7156-7163.

Peffers, K., Tuunanen, T., Rothenberger, M.A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45-77.

Plase, D. (2017). A systematic review of SQL-on-hadoop by using compact data formats. *Baltic Journal of Modern Computing, 5*(2).

Polato, I., Ré, R., Goldman, A., & Kon, F. (2014). A comprehensive view of hadoop research—A systematic literature review. *Journal of Network and Computer Applications, 46*, 1-25.

http://www.tutorhelpdesk.com/homeworkhelp/Computer-Science-/Reverse-Engineering-Process-Assignment-Help.html

Saggi, M.K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing and Management, 54*(5), 758-790.

Sagiroglu, S., & Sinanc, D. (2013, May). Big data: A review. In Collaboration Technologies and Systems (CTS), 2013 International Conference on, 42-47. IEEE.

Sang, G.M., Xu, L., & de Vrieze, P. (2016). A reference architecture for big data systems. In Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on, 370-375. IEEE.

Saraladevi, B., Pazhaniraja, N., Paul, P.V., Basha, M.S.S., & Dhavachelvan, P. (2015). Big data and hadoop-a study in security perspective. *Procedia Computer Science, 50*, 596-601.

Sawant, N., & Shah, H. (2014). Big data application architecture Q&A: A problem-solution approach. Apress.

Singh, H., & Bawa, S. (2018). Scalable metadata management techniques for ultra-large distributed storage systems A systematic review. ACM Computing Surveys (CSUR), *51*(4), 1-37.

Sivarajah, U., Kamal, M.M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research, 70*, 263-286.

Sun, S., Cegielski, C.G., Jia, L., & Hall, D.J. (2018). Understanding the factors affecting the organizational adoption of big data. *The Journal of Computer Information Systems, 58*(3), 193-203.

Vavilapalli, V., Murthy, A., Douglas, C., Agarwal, S., Konar, M., Evans, R., . . . & Baldeschwieler, E. (2013). Apache hadoop YARN: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing, 5*, 1-16.

Veiga, J., Exposito, R.R., Pardo, X.C., Taboada, G.L., & Tourifio, J. (2016). Performance evaluation of big data frameworks for large-scale data analytics. Paper presented at *IEEE International Conference*, 424-431. at

White, T. (2015). Hadoop: The definitive guide 4d ed. " O'Reilly Media, Inc.".

Vaishnavi V., & Kuechler W. (2015). Design science research methods and patterns: innovating information and communication technology [2nd ed]. CRC Press, Boca Raton, FL, 2015. 415. (ISBN 978-1-498715-25-6).

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.

Yaqoob, I., Hashem, I.A.T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N.B., & Datavetenskap. (2016). Big data: From beginning to future. *International Journal of Information Management, 36*(6), 1231-1247.

Zhang, H., Chen, G., Ooi, B.C., Tan, K., & Zhang, M. (2015). *In-memory big data management and processing: A survey*. IEEE Transactions on Knowledge and Data Engineering, *27*(7), 1920-1948.

Received: 27-Dec-2021, Manuscript No. JMIDS-21-9508; **Editor assigned:** 29-Dec-2021, PreQC No. JMIDS-21-9508(PQ); **Reviewed:** 07-Jan-2022, QC No. JMIDS-21-9508; **Revised:** 20-Jan-2022, Manuscript No. JMIDS-21-9508(R); **Published:** 27-Jan-2022

Citation Information: Sharma, R.S., Wingreen, S.C., & Janarthanan, S.B.T. (2022). Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-19.