TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

A Big Data Processing Framework Extended with Data Sets Management

Tae-Hyung Kim[1], Seo-Young Noh[*2]
[1]Principal Engineer, Samsung Research, Samsung Electronics, Seoul, Republic of Korea ,
[*2]Assistant Professor, Department of Computer Science, Chungbuk National University, Cheongju, Republic of Korea.
thkim4u@gmail.com[1], rsyoung@cbnu.ac.kr[*2]

Article Info
Volume 83
Page Number: 4630 - 4637
Publication Issue:
March - April 2020

Abstract
Background/Objectives: Big data deals with massive, compound, diverse data sets. its characteristics are usually denoted using multiple words starting with a letter "V" in industrial fields and academic comminutes. The V characteristics make it extremely difficult for a conventional software system and traditional databases to effectively process and manage big data.

Methods/Statistical analysis: This paper proposes the data sets management approach for dealing with the critical data sets, and presents a generic processing framework for big data and discusses how its internal stages are related to the six V characteristics of big data and quality attributes.

Findings: The purpose of processing big data is to extract the information or generate deliverables valuable to stakeholder and customers using the specific data sets that need to be regularly monitored and updated. For this purpose, the maintenance method for reserve and revise those important data sets used on big data processing are integrated in order to help them slowly aged and keep pace with the rapid and frequent changes of big data.

Improvements/Applications: The big data processing framework is extended with the data sets management methods, which contributes to increase understandability and maintainability of big data itself as well as the design and development of big data processing systems.

Keywords: *Big Data Characteristics, Big Data Processing Framework, Data Aging, Data Recalling, Data Reassessing.*

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 26 March 2020*

1. Introduction

Big data is composed of enormous, complex and heterogeneous data sets. In many researches and industrial fields, the characteristics of big data are represented using the three to more than ten words starting with 'V' [1-5, 7, 15]. Among those V words applied to big data, we think that the five V words, volume, velocity, variety, veracity and variability, are sufficient to describe the characteristics of big data. Firstly, volume is the tremendously huge size of big data that nowadays is larger than petabytes or exabytes. Secondly, velocity means the speed of big data that is produced or changed significantly faster, often in real-time. Thirdly, variety is the forms of big data of which type is basically unstructured and unorganized. It can be used to describe the organization of the data that can be generally categorized in three types: structured, semi-structured, or unstructured [5]. Besides, big data are roughly classified based on data types such as online network, mobile and IoT data, spatial temporal data, streaming and real-time and so on

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

[6]. Fourthly, veracity denotes uncertainty of big data and asks whether the sources of data is realizable and confidence of data is high during its processing. Lastly, variability that is related to consistency of big data describes a multitude of data dimensions that appears during the analysis of big data and makes it difficult to visualize its outcome. In addition to the above five V words that describe the internal characteristics of big data, the most important thing to be excavated from big data and delivered to its stakeholders or customers is value. Note that value hidden in big data is a kind of an objective or a result that can be obtained through big data processing.

The bigness of data leads various issues such as the processing time and cost that increase in proportional to its size and complicatedness. Fundamentally, big data should be collected and analyzed according to the intended purpose and the needs of specific stakeholders or customers. The data sets used during big data processing are tend to be not the center of attention, but it could be useful sometime later when the regression tests are required. Especially, the analysis outcome can be implemented in the form of code and executed as a software service. In this case, this software is very necessary to be checked up regularly and upgraded if necessary. At this time, the big data which the data sets are selected from is also required to be updated owing to some of big data's characteristics mentioned above. For this purpose, the data sets management approach that mitigates data aging is proposed in this paper. Then, a generic processing framework for big data is presented and extended with the data sets management methods proposed.

The paper is described as follows: Section 2 proposes an approach to manage the important data sets used to generate a deliverable that is a model or an analysis result useful to stakeholders. In Section 3, our big data processing framework with nine stages is presented. Section 4 discusses the relationships of its internal stages with the V characteristics mentioned above. Section 5 concludes this paper with future work.

2. Data Sets Management Approach

The data sets become stale very quickly in the world of big data. Software aging [8] that occurs due to failure of covering new requirements or catching the change of technologies and trends decreases the value and quality of software. This kind of aging could be cured with software restructuring that transforms its internal structure and keeps its external interfaces to users equivalent or compatible. Particularly, code refactoring reduces the internal complexity of software developed using object-oriented method without changing its behaviors. Software restructuring and code refactoring are expected to make software aging slow down and improve its quality attributes such as maintainability. It may be detected because of its malfunction or increasing maintenance cost that comes from its low understandability. Data aging, however, tends to be unavoidable and unpredictable. For example, it is hardly possible to guarantee that the data source used to crawl raw data will be accessible in the next month or year. Data aging leads value-decreasing so it should be managed. Note that data itself is not allowed be changed or transformed, which means that we cannot apply an almost similar approach used for software aging. Therefore, our proposed approach takes advantage of the characteristics of big data like volume or velocity. As a first step, we define the data set that are relevant to the analysis result or a value. Then, the two methods named data recalling and data reassembling are explained.

2.1 Principal Data Set

Even in big data era, it is meaningful to reserve and track critical data sets used to extricate information from big data. These critical data set that contributes to the particular deliverable is called the principal data set. In other words, a principal data set converges on the deliverable. Reversely, the principal data set could be identified and

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

segregated by tracing backward to the big data repository from the convergence point relevant to its corresponding deliverable. Each data in a principal data set is always related to a specific time when it is created or collected by a data processing system. Accordingly, the principal data set is described as follows:

Given a data element $E = (t, v)$ where t is either a precise time or a time interval and v is specific data, the data set S is defined as a set of data elements such that $S = \{E_0, \ldots, E_n\}$. The processing result of a data set S generates a deliverable D such that $D = Processing(S)$, which means that the specific data set S is directly relevant to the deliverable D. In this case, the data set S becomes the principal data set P.

A principal data set could be equal to the big data itself when all of the data sets currently stored in its repository are entirely exploited, for example, by an AI classification technique.

2.2 Data Recalling

The data recalling is proposed to reconstitute a principal data set by means of inserting new data elements and deleting unnecessary or outdated data elements. As a principal data set is processed with the data recalling method, its successor should not take any effect on deliverable's value and quality, which could be measured with predefined metrics such as accuracy or precision of which result is within a threshold acceptable to the stakeholders and customers. The data recalling is described using the principal data set as follows:

Given the principal data set $P_i = \{E_0, \ldots, E_k \ldots, E_n\}$ and $0 \leq k \leq n$, the data recalling of P_i generates the principal data set P_{i+1} such that $P_{i+1} = Recalling(P_i)$. The data recalling has the insert and delete operation. The insert operation + is defined as $P_{i+1} = P_i + \{E_{n+1}\}$ where $E_{n+1} = (t_{n+1}, v_{n+1})$ should not be in the P_i, and the time t_{n+1} is later than the time t_n, while the delete operation – is $P_{i+1} = P_i - \{E_k\}$ where E_k is the data element of P_i. The new practical data set P_{i+1} obtained via the data recalling should satisfy the following condition: $Processing(P_{i+1}) = Processing(P_i) \pm \varepsilon$ where ε is a threshold value that does not affect the expected result of the deliverable. For instance, if an analysis method uses accuracy or precision as its quality metric, its result with P_{i+1} should not be changed or decreased when compared with the result of P_i obtained using the same analysis method. Figure 1 depicts the conceptual overview of the data recalling and shows that the insert and delete operation are applied to the principal data set P_i created at the time of t_1.

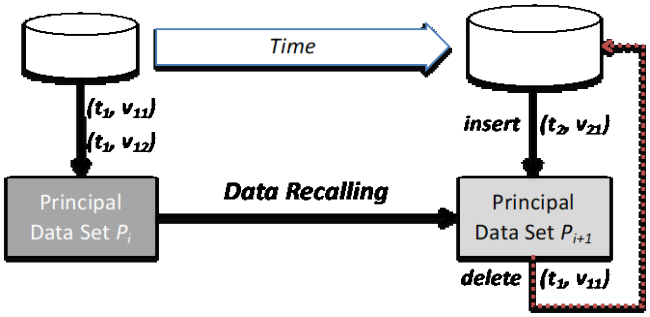

Figure 1. Example of Data Recalling.

2.3 Data Reassembling

As a deliverable is dependent on multiple principle data sets, the data reassessing is proposed to manage the corelated principal data sets, called the principal data group by replacing some of principal data sets, removing the data duplicated across multiple principle data sets or identifying a new one from the principal data sets initially given. As the deliverable is the processing result of multiple data sets such that $D_{multiple} = Processing(P_0, \ldots, P_n)$, its principle data group G is all of the principal data sets included, such that $G = P_0 \oplus \ldots \oplus P_n$. Note that the $\oplus$ operator is not the union operator that removes the data elements duplicated across those multiple data sets. The data reassembling also should fulfill the similar condition such that $Processing(G) = Processing(G') \pm \varepsilon$ where ε is a preset threshold value.

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

The data reassembling method can apply the data recalling to an individual principal data set if necessary. The number of the principal data sets before and after reassessing is irrelevant. The new principal data group obtained using the data reassessing is evaluated by scrutinizing whether the principal data sets are well categorized or classified. Figure 2 shows the example of the data reassembling. In this case, the principal data set P_2 is replaced with a new principal data set P_4 and the data recalling is applied to the principal data set P_3.

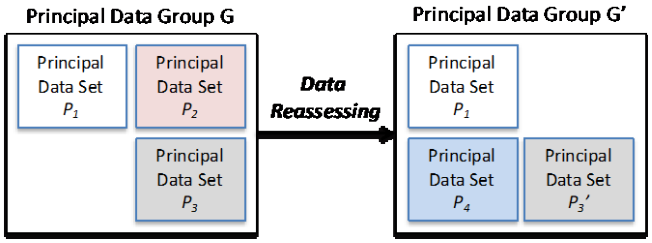

Figure 2. Example of Data Reassembling.

3. An Extended Big Data Processing Framework

The big data processing framework presented in Figure 3 consists of four phases: data collection, data analysis, servitization and data sets management. The objective of each phase is achieved by performing either two or three stages. The stages from the first to the fifth box is similar to the KDD process for discovering knowledge in databases [9]. They are also correspondent to the five steps of the big data analytics application development process that are made of acquisition, presentation, preprocessing, processing and generating/presenting results [3]. A big data processing model with three tiers is proposed from the perspective of the data mining [10]. Actually, those five stages may be enough only for processing big data. Since we want our proposed framework to have ability to manage the principal data set used for creating a service, two phases with four stages are append to extend a basic processing framework for big data.

The first phase gets raw data sets from various data sources and put them into a storage repository. A data lake is the repository or data pool that holds enormous amount of pure raw and unprocessed data sets while a data swamp is the name of a data pool that users are very hard to retrieve and exploit the data efficiently [11]. A data lake becomes a data swamp when it is poorly designed and inadequately maintained. Actually, there is not a clear distinction between the data lake and the data swamp, but variety and veracity can be utilized as primary characteristics in order to distinguish one from the other. The data lake tends to be higher (or more) variety or lower (or less) veracity than the data swamp. The second phase analyzes those raw data sets in the data lake, but a traditional software solution is often inappropriate for this purpose. Note that data processing cost and time tend to be directly proportional to the size of big data treated and strongly dependent on the infrastructure installed and techniques applied [16]. Nowadays, most of big data processing systems are equipped with advanced technologies and AI-based solutions to extract, transform and analyze a variety of data sets. The third phase makes a real service using the result or model obtained after analysis and checks whether a fault occurs or quality of service is decreasing. The data management phase needs to be performed when the service deployed fails to satisfy a level of service criteria or if the analysis outcome such as a model or result value should be modified or reanalyzed in order to reflect the trend of current or up-to-date data.

3.1. Data Collection Phase

The data collection phase has two stages. The gathering stage is responsible for the data sources for raw data sets and a way to crawl them. Some of the raw data sets are moved to the repository with a schema-less database like a modern NoSQL database. If this phase is controlled by some governance model or well-refined rules, the repository contains specific and data already targeted for a predefined purpose and could not be

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

data. For example, if the pictures of flowers are only collected, they are very highly used to classify or categorize types of flowers. In other words, the data source should be controlled to gather the data if a stakeholder has an unambiguous target. Some AI techniques may be applied to in this phase as a kind of filters between the gathering and the storing stage, which can prevent the repository from being a data swamp. However, those filters are possible to take negative effects on data collection task. For instance, they may elude some outlier data that could definitely improve accuracy of the analysis result or quality of the service currently working.

Oftentimes, collecting data is related to the culture of the organization or company that will use the data. The relationship between the five V characteristics, which are the same as the ones explained above, and the culture model of a firm is presented using a theoretical framework based on the organizational culture model [7]. According to the culture model of a firm, this phase may collect only the data necessary. It is interesting that a framework is able to follow the structure and objective of an organization that could restrict the range of data in this collecting phase and fail to retrieve meaningful information.

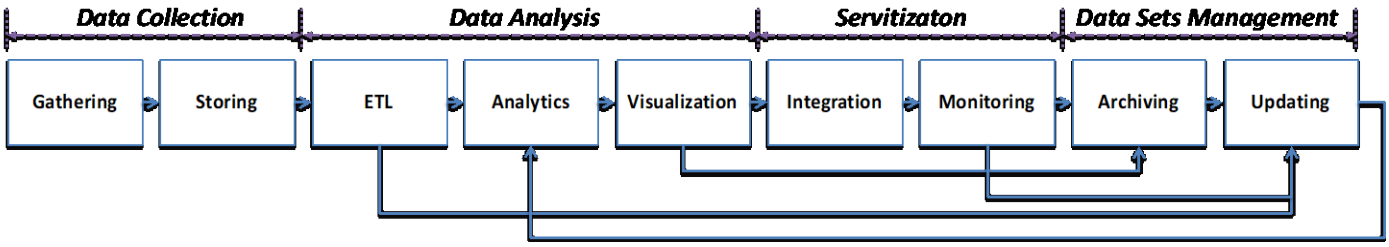

Figure 3. The Extended Big Data Processing Framework.

3.2. Data Analysis Phase

The data analysis phase is comprised of three stages. A set of the collected data is transferred to a disciplined database or data warehouse during the ETL (Extract-Transform-Load) stage and investigated in the analytics stage. Especially, statistical approaches and AI techniques including machine learning and mining unstructured data [12] are aggressively exploited in the during analytics stage in order to make a deliverable that contains abstract models, meaningful results, and so on. The deliverable can be helpful to interpret a current status, give an insight of stakeholders' interests, or forecast the future trends. The analytic stage works well in support of well-managed meta data, strict data governance, relevant data sets and automated process with data cleaning strategy. The multi-level meta data framework to operationalize data governance is presented in [13]. In the visualization stage, the deliverable produced after big data analysis is utilized to help stakeholders or customers understand the information extracted from big data. This stage includes documentation of the deliverable or the information.

3.3. Servitization Phase

The servitization phase is performed only if it is necessary for the visualized information or portions of the deliverable to be integrated into a working system as a form of source code, a concrete service or a microservice. The integration stage follows a traditional software development process or an agile software development method like Scrum [3]. The implementable features are identified based on the information and deliverable obtained from the previous stages. If these updates happen very frequently, this integration stage may involve an exceptionally high maintenance cost.

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

For instance, a machine learning code is at most 5% in a mature system, but clue code is the rest of 95% [14]. The code related to a deliverable tends to be highly coupled with other modules and increase maintenance effort. The monitoring stage regularly checks whether the features embedded into a code or deployed into a service are required to be updated due to some of the V characteristics such as velocity or variability. It will notify user of failing to satisfy the level of service criteria request or exceeding from the threshold value allowed by the analysis method applied in the data the previous phase.

3.4. Servitization Phase

The data sets management phase is designed to mitigate data aging introduced in Section 2. Note that the main purpose of two proposed methods is to retain the principal data sets against data aging. The prerequisite for performing the data recalling or data reassembling is to memorize the principal data sets used for the deliverables. The archiving stage is considered as a kind of the database that can be used to reserve important logs or events generated from the monitoring stage. This archiving stage is not necessary to copy a principal data set from the original data set stored in the repository for the storing stage. It can be designed to store a meta data such as a view that can be either a query set or virtual table with index. However, the principal data set should be copied if the repository storing the original data sets is under the control of a regular cleaning process. The updating stage is the place where the data sets management methods explained the previous section are actively performed. It is able to be activated automatically by the monitoring stage that evaluates the current state of the running service developed through the implementation stage. To reconstruct the principal data set and the principal data group by means of the data recalling and the data reassembling respectively, newer data sets can be directly retrieved from the ETL stage when necessary.

4. Characteristics of the Big Data Processing Framework

In general, most of big data systems cover from the storing stage to the analytics stage or visualization one. They can be built using either open source solutions such as Hadoop, Spark and Strom. Big IT companies prefer to their own proprietary solutions such as Amazon DynamoDB and Google File System [2]. In particular, lambda architecture to is applied to a data processing system that deals with massive data in both batch and stream mode. Out big data processing framework is extended with four stages that are responsible for making the service using the result of the analysis method and managing the data sets used as its input. The V characteristics of big data presented above could be regarded as operational or environmental issues to be considered during the design or development of a big data processing system. Actually, they influence various quality attributes such as availability, security, performance, usability and so on, as well as functional requirements that should be satisfied during the design and development of real big data processing systems. The big data system design method (BDD) devised to help design and develop them is proposed as a combined process model of architecture design with data modeling [15]. Table 1 shows that each stage of the extended big data processing framework presented in Figure 3 is mostly relevant to, but not limited to, one of the V characteristics. For example, the ETL stage primarily is primarily affected by variety because it deals with multiple disparate data types and works with heterogenous analytics stages.

In addition to the traditional quality attributes, scalability needs to be considered to become one of core properties because big data software systems are installed and distributed across multiple locations. For instance, the architectural design of the healthcare informatics system defines tactics, which are elemental and reusable design decisions, for scalability in addition to two basic quality

attributes of performance and availability [17]. The quality attributes mapped to each stage shown in Table 1 is just considered as a representative or an exemplary set. Since repository's capacity closely related with the storing stage is not limitless, it needs to be scalable, which does not mean that other quality attributes like performance and availability is not critical in the storing stage. The main quality attribute for a stage could vary according to the type and purpose of the big data processing system developed. Since privacy becomes more and more important recently, the analytics stage should confirm whether its data sets are well anonymized. Otherwise, this analytics stage is responsible for de-identifying the data sets. Therefore, we choose privacy as its main quality attribute. Moreover, vulnerability may be taken into account as an additional V characteristic of big data if privacy and security are treated as a vital factor in processing big data that contains personal information.

Table 1: The main characteristics and quality attributes of the stages in the big data processing framework

Stage	Input	Output	Mostly related to	Quality Attribute
Gathering	Data sources	Raw data sets	Velocity	Performance
Storing	Simple structure	Data collection	Volume	Scalability
ETL	Rules and data sets	Refined data sets	Variety	Interoperability
Analytics	Hypothesis	Models or Results	Veracity	Privacy
Visualization	Deliverables	Information	Variability	Usability
Integration	Features	Services or APIs	Velocity	Modifiability
Monitoring	Feedbacks	Logs or Events	Value	Availability
Archiving	Data sets used	Meta data	Variability	Consistency
Updating	Meta data	New data sets	Veracity	Testability

5. Conclusion

Big data is not just very large, complex and unorganized data, but the important concept that has been studied in various aspects. Therefore, the six words starting with 'V' are addressed as the major characteristics of big data. The extended big data processing framework focuses on providing services created with the analysis outcome and managing the data sets used as its analysis input, not only on collecting and analyzing big data itself. For this purpose, we define the principal data set contributing to the deliverable obtained as a result of processing big data and propose the two data management methods, data recalling and data reassembling. The main aim of this data sets management approach is to revitalize the principal data sets that become stale over time owing to data aging. The framework we present can help to increase maintainability and understandability of big data processing.

Our research is being directed toward two complementary ways. First, we are expanding our data sets management methods and embedding them into our framework that support. Second, we are customizing our big data processing framework in regard to domain-specific constraints and presenting its application with practical case study, which enables to show how to adjust the V characteristics and quality attributes currently mapped to each stage, identify specific metrics to measure them, and provide the recommendation to

March - April 2020
ISSN: 0193-4120 Page No. 4630 - 4637

solve issues raised during the design and development of big data processing systems.

Acknowledgment

This work has been supported by the Korea Institute of Science and Technology Information(KISTI).

References

[1] Lin YT and Huang SJ. The Design of a Software Engineering Lifecycle Process for Big Data Projects. IT Professional, 2018 Jan/Feb: 45-52.

[2] Karakaya Z. Software Engineering Issues in Big Data Application Development. IEEE 2nd International Conference on Computer Science and Engineering. 2017: 851-855.

[3] Al-Jaroodi J, Hollein B, and Mohamed N. Applying Software Engineering Process for Big Data Analytics Application Development. IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), 2017:1-7.

[4] Madhavji N, Miranskyy A, and Kontogiannis K. Big Picture of Big Data Software Engineering. IEEE/ACM 1st International Workshop on Big Data Software Engineering. 2015:11-14.

[5] Sagiroglu S and Sinanc D. Big Data: A Review. IEEE International Conference on Collaboration Technologies and Systems (CTS). 2013:42-47.

[6] Lv Z, Song H, Basanta-Val P, Steed A, Jo M. Next-Generation Big Data Analysis: State of the Art, Challenges, and Future Research Topics. IEEE Transaction on Industrial Informatics. 2017 Aug;13(4):1891-1899.

[7] Nguyen TL. A Framework for Five Big V's of Big Data and Organizational Culture in Firms. IEEE International Conference on Big Data (Big Data). 2018:5411-5413.

[8] Parnas D. Software Aging. ICSE '94 Proceedings of the 16tth International conference on Software Engineering. 1994:279-287.

[9] Fayyad U, Piatetsky-Shapiro G, Smyth P. From Data Mining to Knowledge Discovery in Databases. AI Magazine. Fall 1996;17(3):37-54.

[10] Wu X, Zhu X, Wu GQ, Ding W. Data Mining with Big Data. IEEE Transactions on Knowledge and Data Engineering. 2014 Jan;26(1):97-107.

[11] Rao V. Data Lakes and Data Swamps. 2018 [updated 2019 March 7; cited 2020 March 4]. Available from https://developer.ibm.com/technologies/analytics/articles/ba-data-becomes-knowledge-2/.

[12] Bavota G. Mining Unstructured Data in Software Repositories: Current and Future Trends. 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). 2016:1-12.

[13] Helvoirt SV, Weigand H. Operational Data Governance via Multi-level Meta Data Management. The 4th Conference on e-Business, e-Services and e-Society (I3E), 2015 Oct:160–172.

[14] Sculley D, Holt G, Golovinm D, Davydov E, Phillips T, Enber D, et. al. Machine Learning: The High Interest Credit Card of Technical Debt. Software Engineering for Machine Learning (NIPS 2014) Workshop, 2014.

[15] Chen H, Kazman R, Haziye S, Hrytsay O. Big Data System Development: An Embedded Case Study with a Global Outsource Firm. IEEE/ACM 1st International Workshop on Big Data Software Engineering, 2015:44-49.

[16] Banerjee S, Cukic B. On the Cost of Mining Very Large Open Source Repositories. IEEE/ACM 1st International Workshop on Big Data Software Engineering. 2015:37-43.

[17] Gorton I, Klein J. Distribution, Data, Deployment: Software Architecture Convergence in Big Data Systems, IEEE Software, 2015 May/June:78-85.