Advances in Social Science, Education and Humanities Research, volume 474
Proceedings of the 2nd International Seminar on Science and Technology (ISSTEC 2019)

Conceptual Approach of Big Data Architecture Mozita Application in Industry 4.0

Aris Puji Widodo[1,*], Djalal ER Riyanto[1], Mukhammad Fakhir Rizal[2]

[1]*Department of Informatics, Faculty of Science and Mathematics, Diponegoro University*
[2]*Master of Information System, School of Post Graduate, Diponegoro University*
[*]*Corresponding author: arispw@gmail.com*

ABSTRACT
In industry 4.0, the growth of data generated from an application will increase in a very large amount of volume and occur massively. The phenomenon of massive data growth with a wide variety of data formats is called the big data. The main problem with big data is that large volumes of data with a variety of formats can provide reliable access speeds to extract data in the form of analytics or data visualization. Likewise, the Mozita application in the industrial era 4.0 will experience a similar problem that occurs in big data. Therefore in this study a conceptual scheme of Mozita application big data architecture conceptual approach using cluster data, batch processing, and stream processing is proposed to provide high reliability in analyzing and visualizing data. The conceptual scheme of the big data architecture developed consists of 3 layers: data processing data collection, and access to analytic and data visualization needs. This big data architecture uses the Hadoop technique as a big data processing framework, with Apache Hive tools, and Cassandra as a NoSQL-based database. This big data architecture can provide high availability and reliability for very large data collection, real-time processing and data access for information extraction in the form of analytics and data visualization. In the application of the new architecture proven to increase speed in data storage by 62.15% and an increase in speed in reading data by 42.38%.
Keywords: *Industry 4.0, Big Data, Analytics, Visualization, and Architecture.*

1. INTRODUCTION

The health industry is currently undergoing a major transformation with a focus on reducing costs and improving the quality and efficiency of health services [1]. The need to provide integrated care services to meet patient health care needs, is a good reason for Health Information Systems that adopt Data Analytics (DA) to ensure reliable and efficient services. Such conditions are becoming increasingly demanding because this enormous volume of health data not only comes from traditional interviews, opname, and medical tests in hospitals or outpatient clinics, but also involves the active role of patients in providing their own data that can be used to telemonitoring, using a health application which can be accessed from anywhere and at any time [2]. In addition, the emergence of new technology followed by detailed information has contributed to the increase in clinical data collected. Process and analyze all of this information, making it possible to identify health patterns that can contribute to healing and preventing illness, in addition to improving patient safety and quality of life. In short, improve the efficiency, quality and savings of the health system. Therefore, new opportunities arise based on the rapid evolution of Big Data technology and the availability of data that can be retrieved [3].

Data management and analytics are very important in health information systems. Data management includes the processes and technologies to obtain, store, prepare and retrieve data for analysis. Analytic, refers to the technique used to analyze and obtain valuable data from Big Data [4]. Investigation shows that organizations that use DA, when managing the decision making process, are more productive and profitable than those who do not [5]. However, it is not clear at this time, to what extent organizations have implemented Analytics, because there are still many challenges in this field [6]. In this case, organizations that intend to increase the use of DA to optimize costs, profitability, productivity and quality must consider strategic investments in this field. Health service organizations are clearly not an exception to this rule. In the field of health care, the Hospital has followed three stages of computerization and data management, namely: data collection, data sharing and (more recent and gradual) data analysis [5]. Collection, storage and analysis of health data are fundamental procedures for providing efficient health services, and their importance increases with the increasing amount of health data that is collected every day [2]. Data sources generate an amount of data every hour, so the data volume increases. Thus, the data storage space will not be

sufficient if the amount of data collected continues to increase, Hadoop Framework which is a tool for analytics of a big data is able to accommodate these needs [7, 8].

In the use of big data required databases that support the storage and stability of each data processing, non-relational databases have become popular in the 2000s, called NoSql by carrying different query languages, the most significant difference between these concepts is that the SQL database is relational and contains foreign keys. In contrast, the NoSql database is irreversible and does not define relationships [9]. In this article the schematic concept will be explained to support the processing of big data from an application called Mozita. The Mozita application is an application used by midwives (cadres) to record and report the nutritional status of toodler in the local health center (Puskesmas). In addition to information on the nutritional status of toddler, the Mozita application can present information that is used by the head of health centers, district health offices and provincial health offices for the purposes of monitoring and decision making in terms of health policy based on the nutritional status of toddler. Therefore in this study, a conceptual scheme of Mozita application big data architecture conceptual approach using cluster data, batch processing, and stream processing is proposed to provide high reliability in analyzing and visualizing data. The conceptual scheme of the big data architecture developed consists of 3 layers: data processing data collection, and access to analytic and data visualization needs.

2. MATERIALS AND METHODS

This paper discusses the literature review as a methodology for conducting research and offers an overview of different types of reviews. Building a research and linking it with existing knowledge is a series of academic research activities, regardless of scientific discipline. Therefore, to do it accurately must be a priority for all academics. However, this task is becoming increasingly complex. The production of knowledge in the field of business research is accelerating at an incredible speed while at the same time remaining fragmented and interdisciplinary. This makes it difficult to keep up with current and up-to-date research, and assess collective evidence in certain fields of research. This is why the literature review as a research method is more relevant than before. Literature review can be broadly described as a more or less systematic way of gathering and synthesizing previous research. Effective and well-conducted reviews as a research method create a strong foundation for advancing knowledge and facilitating theory development. By integrating findings and perspectives from many empirical findings, a literature review can answer research questions with strengths that none of the studies possesses [10].

It can also help provide an overview of the fields in which the research is different and interdisciplinary. In addition, literature review is an excellent way to synthesize research findings to show evidence and to uncover areas where more research is needed, which is an important component for creating theoretical frameworks and building conceptual models. However, the traditional way to describe and describe the literature is often inaccurate and not carried out systematically. This results in a lack of knowledge about what is actually said by a collection of studies that have been or what is the clue of the object of research. As a result, there is a big chance that the authors build their research based on false assumptions. When researchers are selective from the evidence that builds their research, ignoring research that points in another direction, serious problems can be faced. In addition, even when this review methodology is valid, there are often problems with what constitutes a good contribution [10].

Of course, there are already some guidelines for conducting literature reviews that suggest various types of reviews, such as narrative or integrative reviews, systematic reviews, and meta-analyzes or integrative reviews. By building and synthesizing various types of literature reviews, this paper takes several views to get the concept of a big data architecture to be applied in the Mozita Application.

In research conducted by Jie Song, et al. by using OLAP system for big data systems there is still a shortage of not being able to group data and data distribution at once, must use pivot operations using axes (rows and columns) on the dimensions that were previously initiated [11].

In this research, we will use Apache Hadoop as a framework that can help the processing of big data. Hadoop manages large amounts of log data by breaking files into blocks and distributing them to Hadoop cluster nodes, then by using Apache Hive as a data warehouse infrastructure built on Hadoop for summation, queries, and data analysis. It follows the same strategy for computing by breaking down jobs into a number of smaller tasks. Hadoop supports searching and retrieving log data faster than traditional database systems and is error-tolerant [12, 13]. While the database uses Cassandra which has a high value on energy efficiency because of the large volume of data that needs to be stored, questioned, updated, and analyzed [14].

The final step in this research method is performance evaluation by comparing before and after the development of a new architecture. Performance evaluation is done by recording response time when dumping data and displaying data with some variation in the amount of data.

3. RESULTS AND DISCUSSION

3.1. Result

Currently, there are many architectural definitions, emphasizing various aspects and components. The architecture component describes a fundamental aspect of service delivery, the component itself as a functional sub-system, which consists of a series of steps and activities using platforms, tools, and services [15]. In the design, mapped into three main functions contained in Table 1.

Table 1. The main function of architecture

No	Description
1	Data is obtained and converted to a format that is capable of processing by Apache Hive.
2	Data can be sent from data sources to HDFS (The Hadoop Distributed File System)
3	Apache Hive is able to query with data from the Mozita Application stored in HDFS (using Cassandra DB)

HDFS functions as a distributed data storage, then the main thing is Apache Hive and Cassandra DB which will be mutually sustainable in processing data stored in HDFS. In terms of system design, this system consists of various sub-systems which will eventually be integrated. In the architecture that the researchers created for data storage using a Solid State Drive (SSD), the use of SSD in Hadoop has been extensively tested as a pseudo distributed mode and as a medium to large scale cluster, these experiments were conducted on Hadoop benchmark workloads and the general observation was that I/O intensive, shuffle heavy jobs show an improvement in the performance [16].

Here is a picture of the design concept of this system.

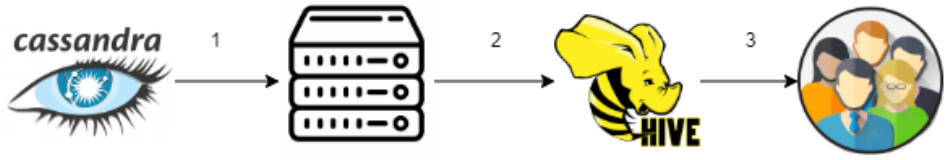


Figure 1. Illustration of system design

The following is an explanation of the processes that occur in the picture.

1. The process of conversion and dump from Cassandra DB (NoSQL) to JSON file, compression to .zip format, to sending data to S3. At the time of this process, the system will simultaneously record the time and speed when starting to dump when finished dumping, record the load average, the memory usage of the server.
2. Data that has been downloaded is entered into HDFS, then recorded the length of the process of data storage and load, memory usage from the server. In MapReduce, the job is processed in two main phases (map and reduce type tasks). Once the mapping is done, the job is shuffled and partitioned and then send to reducer for further processing. So in the MapReduce framework, the tasks are executed in the following sequence:
    - Map Phase- retrieves input data and generates key-value pairs using the Input Format mapper. These key-value pairs are processed according to the map function output so that the generated output is stored on local disks and status and the progress is communicated to the master node (task tracker).
    - Shuffle and sort phase- the input to the reducer is sorted according to the key. This sorting is done and transferring of map outputs as input to the next phase i.e. to the reducer is done by the shuffle phase.
    - Reduce phase- the sorted key is processed and the output file is stored in HDFS.

Proper scheduling will improve cluster's performance, by default, Hadoop has three scheduling algorithms- First come first serve or first in first out (FIFO), capacity scheduler, and fair scheduler [17].

In the second process MapReduce is implemented, where one of the most widely used programming models for analyzing large-scale datasets. Apache Hadoop provides execution support for the MapReduce programming model. Default setting of number of Mappers and Reducers are 12 and 4 respectively; it is explicitly stated whenever it is changed.

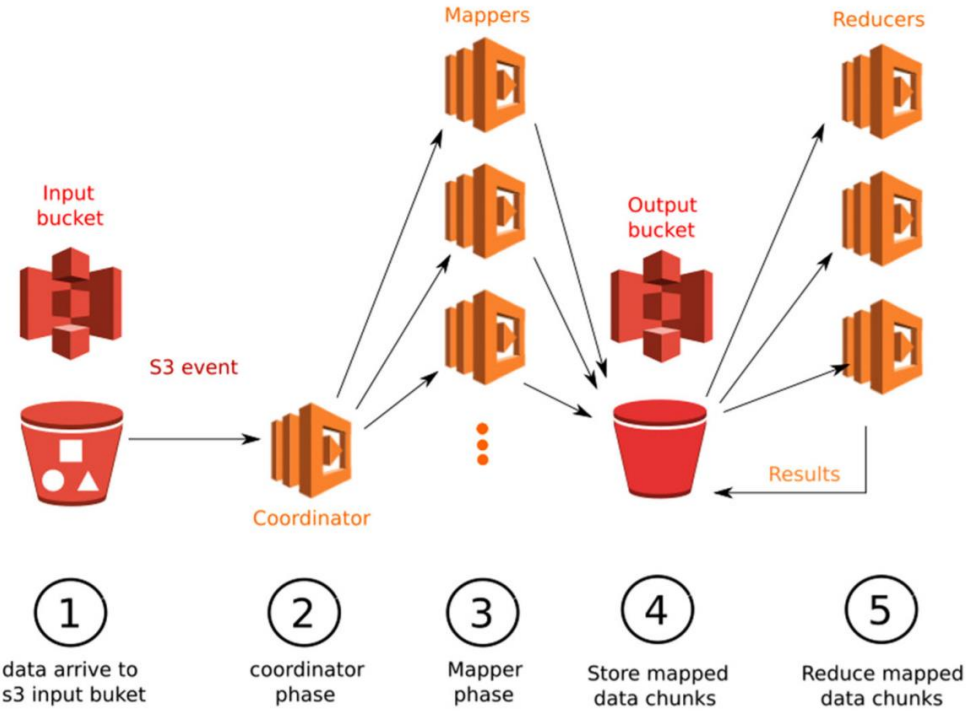


Figure 2. Architecture to support the use of MapReduce [18]

Figure 2 shows an architectural diagram. The architecture is fully composed by the Cloud service, it consists of three functional groups: Coordinator, mapmaker and reduction. In addition, two S3 buckets are needed, one for data input and another for collecting post-processing data generated as output. The workflow execution of MapReduce jobs starts when the dataset is uploaded in the "input" S3 bucket. This causes the S3 event to enable requests to the coordinator function, which calculates the optimal size of the data partition which divides the total dataset size into as many chunks as the user indicated (N) provided the mapper function can store in memory the amount of data in the allocated RAM. If the mapping does not have enough RAM, the coordinator function will recalculate the number of matching pieces in each mapper. To calculate the size of the pieces of data that are processed by each mapper, the coordinator function performs the following process. First, it calculates the possible size of each piece of data that divides the file size for processing by the user-specified number of mappings.

Next, each mapper downloads some data to do the mapping phase in parallel with other requests. Then, the mapper function will sort the data and divide the mapped results in chunks (mapped chunks). The final step is to ensure a good distribution of all pieces of data between S3 partitions, thereby increasing the performance of data access in S3.

3. Apache Hive queries the files in the HDFS. Load, memory usage of the server when doing this process will be recorded, the duration of the query will be recorded in the Apache Hive's own log.

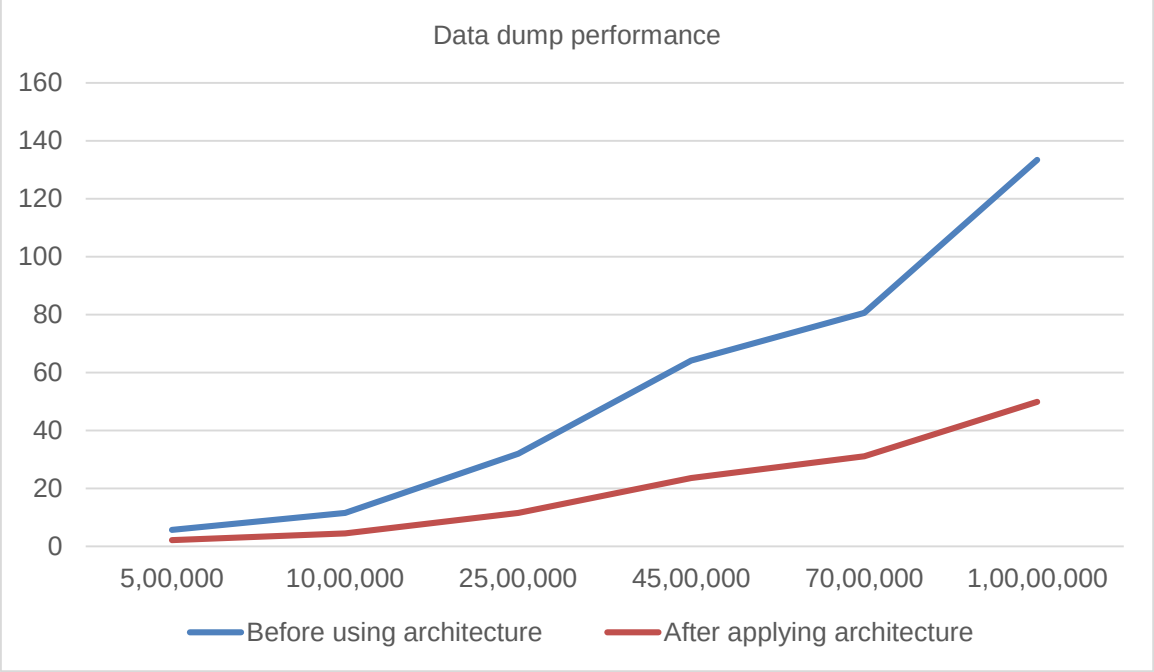
3.2. Qualitative Analysis

Tests conducted in this study consisted of 2 tests. Testing dumping data to the database is in the form of adding toddlers data shown in Table 2 and retrieving data from the database shown in Table 3.

Table 2. Data dump performance

Amount of Data	Before using architecture (x)	After applying architecture (y)	Increase
500.000	5.721 s	2.213 s	61.32%
1.000.000	11.515 s	4.528 s	60.68%
2.500.000	31.974 s	11.597 s	63.73%
4.500.000	64.117 s	23.612 s	63.17%

7.000.000	80.648 s	31.141 s	61.39%
10.000.000	133.451 s	49.901 s	62.61%



From the data in Table 2 we get the results of an average increase of 62.15%, with the calculation of the increase as follows:

$$\text{Increase} = \frac{x - y}{x}\ x\ 100\%$$

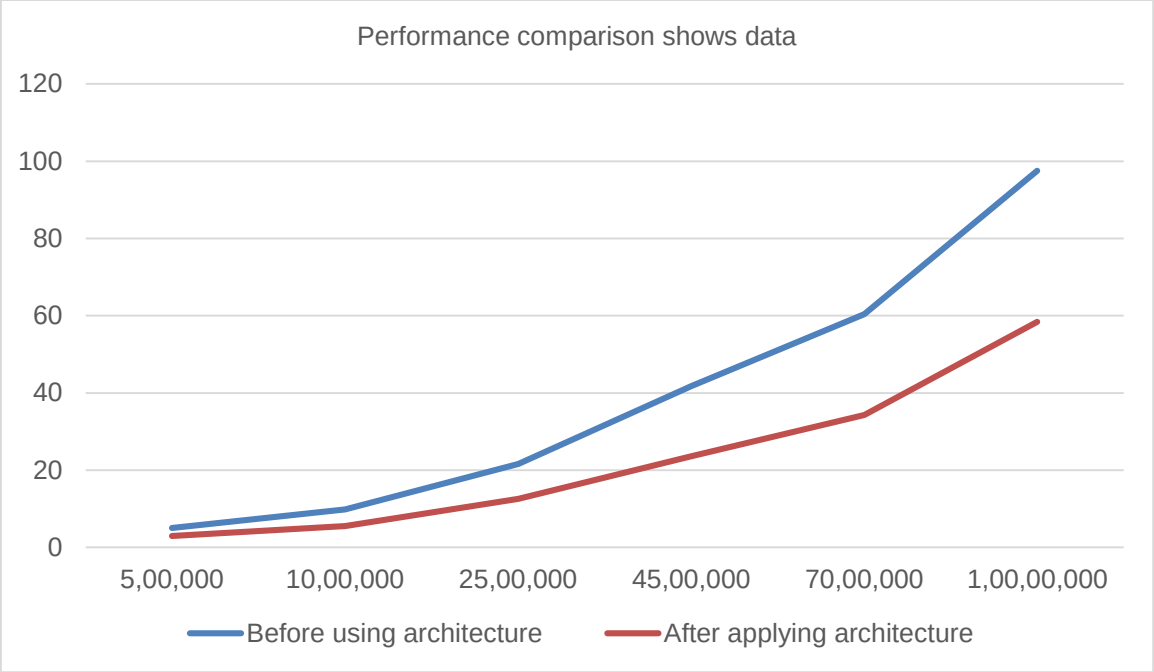
As, x = time dump before using architecture
y = time dump after using architecture

From the test that the length of time the dump is quite linear, this is quite good when in the future can estimate the length of the dump. One of the things that took a long time to dump was a query reading the database list at the beginning of the process. Second is when data is stored, where data is stored in several databases.

In the next test, nutrition data collection for children under five is taken from several databases at once, and the data collection process will be compared before applying this architecture, which can be seen in Table 3.

Table 3. Performance comparison shows data

Amount of Data	Before using architecture (x)	After applying architecture (y)	Increase
500.000	5.033 s	2.919 s	42.01%
1.000.000	9.848 s	5.527 s	43.87%
2.500.000	21.623 s	12.598 s	41.74%
4.500.000	41.711 s	23.625 s	43.36%
7.000.000	60.361 s	34.278 s	43.21%
10.000.000	97.531 s	58.397 s	40.12%



In Table 3 we can see an increase in speed, although as more data is taken the average speed decreases, it is not significant. With the same calculation formula with data dump performance testing,

$$Incrase = \frac{x - y}{x}\ x\ 100\%$$

As, x = respone time before using architecture
y = respone time after using architecture

The conclusion is an increase from before applying the big data architecture to after there was an increase of about 42.38%.

4. CONCLUSION

This paper presents the design, implementation, and evaluation of architecture to accommodate the issue of big data and the demands of industrial development 4.0 (especially in the field of health information systems). This architecture was built using the Hadoop framework with its core in the form of the Hadoop Distributed File System (HDFS) where data and files are stored and Map Reduce to do data mining and other data processing of files or data stored in HDFS, MapReduce handles a huge amount of data gracefully regardless of the hardware restrictions. Apache Hive is used to make it easier to create and run Map Reduce, so it will be easier to create and run queries. While the database used here is Cassandra which is highly scalable and is designed to manage structured data with very large capacity (Big Data) spread across many servers, Cassandra is one of the implementations of NoSQL. We compare system performance from before and after implementing the new architecture with an average increase of 62.15% in terms of dumping data to the database, and an average increase of 42.38% in terms of reading data.

ACKNOWLEDGMENT

This work was supported by Research Funding Sciences and Mathematics Faculty Diponegoro University 2019.

REFERENCES

[1] Wu, J. H., Kao, H. Y., Sambamurthy, V. (2016). The Integration Effort And E-Health Compatibility Effect and the Mediating Role of E-Health Synergy on Hospital Performance. International Journal of Information Management. 36(6): 1288-1300.

[2] Gisele, R. (2016). Big Data in Healthcare. Journal of Healthcare Communications ISSN 2472-1654. 1(4): 33.

[3] Raguseo, E. (2018). Big Data Technologies: An Empirical Investigation on Their Adoption, Benefits and Risks for Companies. International Journal of Information Management. 38(1): 187-195.

[4] Gandomi, A., Haider, M. (2015). Beyond the Hype: Big Data Concepts, Methods, and Analytics. International Journal of Information Management. 35(2): 137-144.

[5] Carvalho, J. V., Rocha, Á., Vasconcelos, J., Abreu, A. (2019). A Health Data Analytics Maturity

Model for Hospitals Information Systems. International Journal of Information Management. 46: 278-285.

[6] Lismont, J., Vanthienen, J., Baesens, B., Lemahieu, W. (2017). Defining analytics maturity indicators: A Survey Approach. International Journal of Information Management. 37(3): 114-124.

[7] Bhathal, G. S., Singh, A. (2019). Big data: Hadoop Framework Vulnerabilities, Security Issues and Attacks. Array. 1: 100002.

[8] Roman, Č., Michal, K. (2019). Comparison of Query Performance in Relational a Non-relation Databases. Transportation Research Procedia. 40: 170-177.

[9] Snyder, H. (2019). Literature Review as a Research Methodology: An Overview and Guidelines. Journal of Business Research, Volume 104: 333-339.

[10] Song, J., Guo, C., Wang, Z., Zhang, Y., Yu, G., Pierson, J. M. (2015). HaoLap: A Hadoop Based OLAP System For Big Data. Journal of Systems and Software. 102: 167-181.

[11] Mavridis, I., Karatza, H. (2017). Performance Evaluation of Cloud-based Log File Analysis with Apache Hadoop and Apache Spark. Journal of Systems and Software. 125: 133-151.

[12] Rodger, J. A. (2015). Discovery of Medical Big Data Analytics: Improving the Prediction Of Traumatic Brain Injury Survival Rates By Data Mining Patient Informatics Processing Software Hybrid Hadoop Hive. Informatics in Medicine Unlocked. 1: 17-26.

[13] Mahajan, D., Blakeney, C., Zong, Z. (2019). Improving the Energy Efficiency of Relational and Nosql Databases Via Query Optimizations. Sustainable Computing: Informatics and Systems. 22: 120-133.

[14] Lnenicka, M., Komarkova, J. (2019). Developing a Government Enterprise Architecture Framework to Support the Requirements of Big and Open Linked Data with the Use of Cloud Computing. International Journal of Information Management. 46: 124-141.

[15] Giménez-Alventosa, V., Moltó, G., Caballer, M. (2019). A Framework and a Performance Assessment for Serverless MapReduce on AWS Lambda. Future Generation Computer Systems. 97: 259-274