ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

Any Where to Any Where: A Data Lake Architecture Using KAFKA

Rahul Kumar * , Dr. Subodh Kumar, Dr. Saurabh Gupta

Research Scholar Department of Computer Applications, IET Mangalayatan University Aligarh-U.P. (India)
Associate Professor Department of Computer Applications, IET Mangalayatan University Aligarh-U.P. (India)
Deputy Director General National Informatics Centre, Delhi. (India)

ABSTRACT*:* Big data explore new opportunities to modern era for discovering new information and knowledge for better understanding and rapid decision making. In rapid growing industries lots of data is generating from different-different sources but to get proper meaningful information there is a demand of market to take all the generated data on one place. Researcher focuses on these types of issues and designs open source architecture to integrate data from heterogeneous data sources on one place. The main focus on this study is sources of data may be any type any technology and target of data sources may be any type. The architecture will work as plug in and play, means it will support any type of sources and any type of target data stores. As per background studies there is no open source architecture available to support complete heterogeneous data sources. The framework is designed by using open sources tools and techniques. Additionally with the integration approach a search algorithm also proposed to search record with minimum time. A well defined analytics approach also proposed to convert Data Lake to meaningful information. Using proposed architecture data analytics has been significantly improved over the other contemporary approaches.

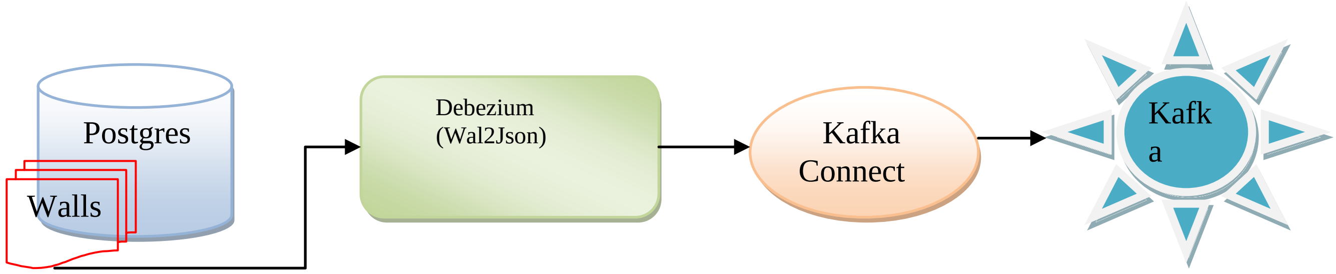
KEYWORDS: Kafka, Zookeeper, Debezium, Elastic search, Kafka-Connect
.

1. INTRODUCTION

The massive size of both types of i.e. structured and unstructured data which is high in volume and difficult to process by using existing traditional computing technique is known as big data. The ample of new opportunities for discovering new information and knowledge are available in the field of bid data analytics in the modern era. The work in the field of bid data is infatuated by the very large amounts of high-dimensional or unstructured data which is continuously generated and stored with a much cheaper cost for further use. Now days the size of big data is increasing day by day with massive rate in every sector and the term big data became very popular now days due to large amount of information's are added every second. This huge amount of data is further need to be processed for efficient decision making and strategic planning.

Now days advanced big data analytics techniques are available which are used to analyze the enormous and diverse data sets that include different types of data such as structured/unstructured and streaming/batch with different size from terabytes to zettabytes. Big data is a term applied to data sets whose size or type is beyond the ability of traditional relational databases to capture, manage, and process the data with low-latency.

In this research Debezium play very vital role to capture changes in source database. Debezium is a distributed platform that turns your existing databases into event streams, so applications can see and respond immediately to each row-level change in the databases. Debezium is built on top of Apache Kafka and provides Kafka Connect compatible connectors that monitor specific database management systems. Debezium records the history of data changes in Kafka logs, from where your application consumes them. This makes it possible for your application to easily consume all of the events correctly and completely. Even if your application stops (or crashes), upon restart it will start consuming the events where it left off so it misses nothing.



Debezium takes care to notify the changes in postgres to debezium client

Debezium writes kafka topics by using Kafka connector

Only committed data to be read from Walls

Figure: Debezium Connector

Till now very limited techniques have been developed to process large set of data. However engineers of this field are scrupulously developing every day a new solution to handle the massive data. Till now there is no standard technique or architecture available to handle both structured and unstructured data in a dexterous way.

Hence in the present study an attempt has been made by the researchers to design and develop a proficient solution to tackle the problems of both structured and unstructured data.

The aim of this architecture is to knob the integration of structure and unstructured data and to minimize the processing time limit and maximize the reusability of data. Designed architecture is used to successfully migrate the data from different-different sources to different-different targets ***(any where to any where)***.

In this architecture we are taking Postgres database as source and Elastic search as target database.

In suggested solution one can also take other sources and targets of databases like File system, Mysql and No sql as sources and Hadoop, Elastic search and NoSql as targeted database.

The organization of present paper is in six different sections namely section-1: Introduction, Section-2: related work, section-3: Data Integration Challenges of Big Data, section-4: Flow of Data Integration in Big Data, section-5: Conclusion and lastly paper ended with references and authors brief profile.

II. RELATED WORK

In present time, the digital universe comprise all structured and unstructured data spanning from videos, movies, surveillance video, photographs, data recorded through sensors, connected devices etc. and it is expected to increase up to 44 zettabytes by the 2020 [1]. Till now numbers of studies have been conducted in the field of big data at national and international platform. This section covers the related work conducted so far in this field as follows:

For continuing the part of literature review I have studied from official's site of tools that have I used in the proposed architecture.

LXC (Linux Containers) was the first, most complete implementation of Linux container manager. It was implemented in 2008 using cgroups and Linux namespaces, and it works on a single Linux kernel without requiring any patches.

Cloud Foundry started Warden in 2011, using LXC in the early stages and later replacing it with its own implementation. Warden can isolate environments on any operating system, running as a daemon and providing an API for container management. It developed a client-server model to manage a collection of containers across multiple hosts, and Warden includes a service to manage cgroups, namespaces and the process life cycle.

Included in 2013 as an open-source version of Google's container stack, providing Linux application containers. Applications can be made "container aware," creating and managing their own sub containers. Active deployment in LMCTFY stopped in 2015 after Google started contributing core LMCTFY concepts to lib container, which is now part of the Open Container Foundation. All those iterations had their adopters and devotees, but when Docker emerged in 2013, containers exploded in popularity. It's no coincidence the growth of Docker and container use goes hand-in-hand. Just as Warden did, Docker also used LXC in its initial stages and later replaced that container manager with its own library, libcontainer. But I've no doubt that Docker separated itself from the pack by offering an entire ecosystem for container management. With Docker, developers can create and run application containers quickly. And with the release of Docker Hub, developers can download and run application containers even faster.

Kubernetes (Greek for "helmsman" or "pilot") was founded by Joe Beda, Brendan Burns and Craig McLuckie, was quickly joined by other Google engineers including Brian Grant and Tim Hockin, and was first announced by Google in mid-2014. Its development and design are heavily influenced by Google's Borg system, and many of the top contributors to the project previously worked on Borg. The original codename for Kubernetes within Google was Project Seven, a reference to a Star Trek character that is a 'friendlier' Borg. Kubernetes v1.0 was released on July 21, 2015. Along with the Kubernetes v1.0 release, Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF) and offered Kubernetes as a seed technology.

Kubernetes (commonly referred to as "K8s") is an open-source system for automating deployment, scaling and management of containerized applications that was originally designed by Google and donated to the Cloud Native Computing Foundation. It aims to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts". It supports a range of container tools, including Dockers.

Apache Kudu is a free and open source column-oriented data store of the Apache Hadoop ecosystem. It is compatible with most of the data processing frameworks in the Hadoop environment. It provides a completes Hadoop's storage layer to enable fast analytics on fast data.

Debezium is an open source project that provides a low latency data streaming platform for change data capture (CDC). You setup and configure Debezium to monitor your databases, and then your applications consume events for each row-level change made to the database. Only committed changes are visible, so your application doesn't have to worry about transactions or changes that are rolled back. Debezium provides a single model of all change events, so your application does not have to worry about the intricacies of each kind of database management system. Additionally, since Debezium records the history of data changes in durable, replicated logs, your application can be stopped and restarted at any time, and it will be able to consume all of the events it missed while it was not running, ensuring that all events are processed correctly and completely.

From lastly three to four years, the amount of data streams has not stopped to increase in many fields such as financial applications, network monitoring, sensor networks and web log mining [10]. Indeed, the Internet of Things accelerated the development of stream computing through the fast expansion of sensors deployed at geographically distributed locations [11]. Large Internet companies also shifted their workloads towards the stream model: for example, Facebook has to handle 106 events/s within a latency between 10 and 30s for advertisement purposes, which represents a data rate of 9GB/s at peak [10].

Le Paul Noac'h , Costan Alexandru Inria,Luc Bouge´ ENS[12], emphasized how to detect the bottlenecks of of Apache Kafka and how to fine tune its deployment. Researcher focus on optimizing the Big Data processing architecture for a given specific use-case. As the mechanisms of ingestion are better understood.

Much streaming data analysis using deep learning has been studied in recent years. Algorithms and architectures for high-speed and high-accuracy deep learning are being studied [13][14][15]. However, these are premised on executing streaming data on a single computer. This research is different in that we consider the total analysis throughput, taking into account the data transmission between different-different sources to final point. In addition, because stream processing is performed using Spark Streaming, it is possible to easily perform extensions using the functions of Spark. Spark Streaming is applied to various technologies. The paper [16] proposes DINAMITE, which is a tool kit that provides analysis tools implemented by Spark Streaming. The analysis tools of DINAMITE measure all memory access with advanced debugging information and help programmers to identify memory bottlenecks. Chen and Bordbar use Spark Streaming as a solution to issues such as speed, scalability, and fault tolerance of rule-based systems that are currently used [17].

Ayae Ichinose, 2 Hitotsubashi, Chiyoda-ku , Otsuka, Bunkyo-ku [18] propose a video analysis framework that collects videos from multiple cameras and analyzes them using Apache Kafka and Apache Spark Streaming. In addition, it is confirmed that the number of cores is needed to consider for the efficient cluster configuration, and that the network bandwidth between the nodes becomes a bottleneck as the amount of data and the number of components increase. In this study the author mainly focus on particular dataset not generic dataset.

III. DATA INTEGRATION CHALLENGES OF BIG DATA

Every organization wants high availability information based on generated dynamic data from various sources and in various formats the there is big challenge of data integration on central place to process and generate predictive information. To integrate the data from various sources and from various formats to central place there is architecture required which can provide a common solution and can fit any requirements. There is another challenge to streaming continuously and integrate legacy data which may be in different formats.

The following key challenges of data integration

1) Volume of data is very huge.
2) Format of data (structured and unstructured)
3) Heterogeneous data input sources.
4) Heterogeneous data output sources.
5) Maintenance of data (Live Streaming).
6) Incremental backup.

To overcome these challenges there is a need to develop framework or any architecture. There are many open source tools available in market to integrate big data like apache Sqoop, apache Kafka etc. With the help of these tools authors developed architecture to integrate and streaming big data from different-different data sources to different-different target sources.

IV. FLOW OF DATA INTEGRATION IN BIG DATA

To migrate different types of data in big data firstly it is necessary to know about the sources of data and its formats. After identification of sources of data, the following steps are to follow:

Step 1: Identify the platform for compellability of selection of tools.
Step 2: Install compatible tools on source and target platform.
Step 3: Start capturing data logs on sources of data.
Step 4: Change compatible format of source data to big data formats.
Step 5: Stream processing in Kafka and make data to target compatible.
Step 6: Store formatted data in big data warehouse.
Step 7: Process and analytics data based on given trends and generates desired reports.
Step 8: Extract data from central repository and provide the client as per requirement using API.

After data integration the next challenge is to stream the data.

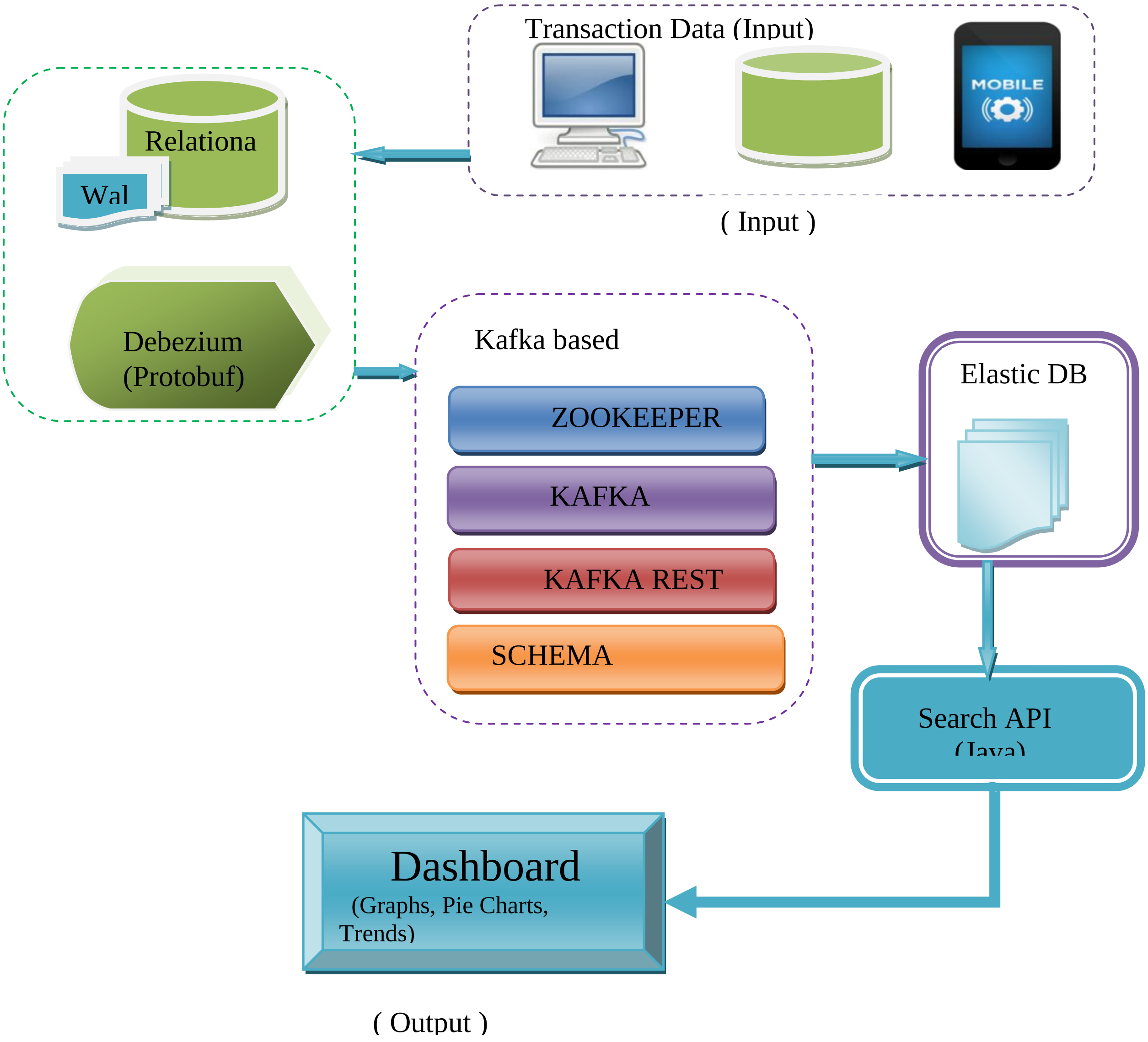


Figure 1: Data Integration Architecture of Big Data

In above figure-1, the data integration architecture of big data with help of open source tools has been laid out. In this architecture the data will continue replicate from online data transactions including legacy data to central repository.

The main concept of this architecture is to minimize the time limit and maximize the reusability. The target of the given architecture is to successfully migrate the data from different-different sources to different-different targets. In this architecture we are taking Postgres database as source and Elastic search as target database. In this solution we can also take other sources and target of databases like File system, Mysql and No sql as sources and Hadoop, Elastic search and NoSql as targeted database.

V. CONCLUSION

In paper less era data is growing very fast. As Forbes report there are 2.5 quintillion bytes of data created each day at our current pace. Data is collected in different –different formats such as text, images, audio and video (structured and unstructured). To process those data and get meaningful information from it there is required to integrate on one central place. The depended data is generated from heterogeneous sources. Hence data integration is big challenge in current scenario. To ease of data integration process authors are proposed a common solution to integrate data from many sources to one place. In this architecture the main focus on input sources type and output destination type. There is no limitation of input sources type any sources can plug in as input sources and any target data point may become destination data warehouse (any where to any where). In present study take an example to integrate data from Postgres (RDBMS) to Elsticsearch as destination. In this architecture authors used Debezium to read archives and Kafka, zookeeper, Kafka schema for core integration and Elastic search use for central repository. On top of Elsticsearch authors developed a java API to provide the data as user required.

REFERENCES

[1] https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016.
[2] https://docs.docker.com/
[3] https://kubernetes.io/
[4] http://edit.dialogic.com/glossary/kubernetes
[5] https://github.com/kubernetes/kubernetes/
[6] https://www.wired.com/2015/06/google-kubernetes-says-future-cloud-computing/
[7] https://kudu.apache.org
[8] https://debezium.io/
[9] https://github.com/**debezium**/debezium
[10] Lu, Ruirui, Wu, Gang, Xie, Bin and Hu, Jingtong. "Stream Bench: Towards Benchmarking Modern Distributed Stream Computing Frameworks.." Paper presented at the meeting of the UCC, 2014.
[11] Milan ermk, Daniel Tovark, Martin Latovika and Pavel eleda, A Performance Benchmark for NetFlow Data Analysis on Distributed Stream Processing Systems, Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP
[12] Le Paul Noac'h , Costan Alexandru Inria,Luc Bouge´ ENS "A Performance Evaluation of Apache Kafka in Support of Big Data Streaming Applications" 978-1-5386-2715-0/17 2017 IEEE.
[13] L. Qing, Q. Zhaofan, Y. Ting, M. Tao, R. Yong, and L. Jiebo, "Action recognition by learning deep multi-granular spatio-temporal video representation," in Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ser. ICMR '16. New York, NY, USA: ACM, 2016, pp. 159–166
[14] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, "Evaluating two-stream cnn for video classification," in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ser. ICMR '15. New York, NY, USA: ACM, 2015, pp. 435–442
[15] J. Read, F. Perez-Cruz, and A. Bifet, "Deep learning in partiallylabeled data streams," in Proceedings of the 30th Annual ACM Symposium on Applied Computing, ser. SAC '15. New York, NY, USA: ACM, 2015, pp. 954–959
[16] S. Miucin, C. Brady, and A. Fedorova, "End-to-end memory behavior profiling with dinamite," in Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ser. FSE 2016. New York, NY, USA: ACM, 2016, pp. 1042–1046.
[17] Y. Chen and B. Bordbar, "Dress: A rule engine on spark for event stream processing," in Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, ser. BDCAT '16. New York, NY, USA: ACM, 2016, pp. 46–51.
[18] Ayae Ichinose, 2 Hitotsubashi, Chiyoda-ku , Otsuka, Bunkyo-ku "A Study of a Video Analysis Framework Using Kafka and Spark Streaming" 978-1-5386-2715-0/17 2017 IEEE

.AUTHOR'S BIOGRAPHY

Rahul Kumar is Sr. Software Engineer, National Informatics Center New Delhi. He is pursuing his Ph.D. from Mangalayatan University, Aligarh and the Master degree (MCA) had done from there with 89.10%. He is also working as new technology advisor and also working with big data Analytics team at National Transport Project, National Informatics Center HQ Delhi. He had 3+ Year of extensive experience of Java developer.

Dr. Subodh Kumar is Head in Department of Computer Applications, Institute of Engineering and Technology, Mangalayatan University, Aligarh. He has obtained his Ph. D. from Mangalayatan University, Aligarh and the Master degree in computer application from Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow (formerly Uttrapradesh Technical University). He has been meticulously carried out research activities since last 9 years in the field of Mobilie Ad-hoc Network, Big Data, Software Engineering and Database. He has taught various courses at graduate as well as at postgraduate level and supervised number of projects and dissertations of BCA & MCA students. Presently he is supervising five Ph. D scholars in the field of Big Data, Mobile Ad-hoc Network, and Cyber Security. He has published 15 research papers in International Journals of repute, and has been part of many National and International conferences.

Dr. Saurabh Gupta is Deputy Director General, National Informatics Center and Joint Secretory in MHA Govt. of India, and Ph.D. in Computer Engineering, Degree in Management and Law with Certification in Management Development Programmed from IIM, Bangalore and Strategic management from IIM Lucknow. Besides this he has many certification such as Synthetic Aperture Ruder Technology (SAR) from IIT- Bombay, Certification in "Certified

Software Quality Professional” from STQC-Delhi, Certification in ”Internal Quality Auditor for ISO” from STQC , Certification in “Accounts and Administration” from ISTM Delhi and Certification on Decentralized Planning under 73rdAmendment of Constitution in SIRD, Lucknow.