A Novel Framework for Enhancing QoS of Big Data

Dar Masroof Amin[1]
MMICT&BM,
Maharishi Markandeshwar (Deemed to be University),
Mullana, Haryana 133203, India

Dr. Munishwar Rai[2]
MMICT&BM
Maharishi Markandeshwar (Deemed to be University)
Mullana, Haryana 133203, India

***Abstract*—The dire increase in number of devices connected to the internet is making inherent growth in creation of data. The use of data science in research is creating opportunities for better business analytics and generation of future trends. The data is growing with ever increasing rate and the exponential growth of data is creating opportunities for utilizing the same in process of analysis. The techniques and technology in place is not able to cater the needs of growing data on the Internet. The research work presented here provides a novel framework for improving the performance and management of big data clusters. The research proposed provides a detailed aspect how big data can be handled in the respective domains. The prime aim of this research is to formulate and implement an algorithm by testing with different data sets which can make the process of mining and handling big data easy in the organizations. The framework provides optimized results as compared to traditional systems.**

Keywords*—IoT (internet of things); big data; DSDSS (Domain Specific Data Distribution Algorithm); AI (Artificial Intelligence); ML (Machine Learning)

I. Introduction

Big Data terminology is generally applied to the data that grows exponentially and which cannot be accessed by using conventional database systems. The size of data sets involved in big data cannot be handled by traditional software technology and databases. The common tools, storage systems cannot store, process and manage the size of datasets [1]. The size of big data sets is increasing and ranges from few terabytes to petabytes in a lone data set. As a result, various difficulties occur which are related to storage, mining, distribution and analyzing of these big data sets. In current scenario the organizations are using the high definition data to explore hidden patterns that were not known [18]. Therefore analytics of big data uses advanced algorithms and techniques that are applied for carving out hidden data definitions. The big data technologies are replacing the traditional tools for accessing and manipulating the large amount of data created by online systems [31]. The data is gathered in real time manner from exponentially growing big data systems. The social networking application like Twitter analyses the data collected for grammatical corrections and query prediction by using searching algorithm [12]. By using big data techniques, Netflix provides ranked customer services and other user friendly commendations [25]. Similarly, LinkedIn provides services like news feed, skill promotions and mostly the “persons you may know”. The pattern followed by the consumers can be understudied by analyzing the collected data. The future trend can also be predicted by following the same pattern. The monitoring of network traffic can also be deducted from these data values used by the applications [9]. In order to improve the process of manufacturing these big data values can be used for finding out digital displays process [8].

The larger data sets are comparatively more complicated to handle [17]. The objectives can be achieved only when complex and large data sets with real time potential are visualized in practical format. But in order to achieve this real time goal of complex data sets there is need of proposing new frameworks, tools and analytical models. The research presented provides a detailed methodology with corresponding analytical tools for dealing with growing big data related issues. The research starts with presenting a framework with algorithm for solving issues arising out of mining big data.

The prime goal of research presented is to use big data sets in the designed framework which enhances QoS in mining the big data. The rest of the paper is divided as: The Section II gives details about literature behind the proposed research. Section III provides the detailed design of the proposed framework. This part-c of the same section provides an optimization algorithm for enhancing QoS. The algorithm proposed optimizes the flow of big data clusters in the system. The results and related discussion has been covered in Section IV. And Section V provides the conclusion and future scope of the research presented.

II. Literature Review

The researchers have put forth a framework for mining big data [29]. The research presented a theorem which provides Heterogeneous, Autonomous, Complex and Evolving relationships between data for characterizing big data. The theorem proposed was named as HACE theorem. The authors proposed a big data mining platform which is comprised of three layers. The framework has mining algorithms along with application knowledge and semantics. The big data varies from traditional approaches to unstructured and real time structured data [11, 7]. Chen et al. provided a detailed survey of big data ontologies [6]. The work gives details about principles of design, techniques, challenges and opportunities provide by big data. M. Chen and other researchers provided an inclusive analysis of big data [16]. The research conceals applications, data collection, storage and technologies in addition to its use in future. The research also covers how to adopted new techniques, various platforms of information systems and taxonomy of architectures. Cuzzocrea et al. provided details about privacy in big data posting and also discussed big data over OLAP (Online Analytical Processing) agenda [2]. The E.Begoli surveyed platforms and architectures which were mean for data analysis in large scales [10]. A validated architecture for big data has been proposed by Zhong et al. for supporting high speed queries and frequently coming updates [27]. The system proposed contains distributed processing and in-memory data containing system for analysis of tasks. To provide a separate data generation system and semantic analysis, Cuesta proposed architecture (SOLID) which is having various tiers for separating and a big data management system [5]. The predictive analysis of archive data and structured real-time data has been proposed in [20]. The authors proposed an architecture that is service-related for domains used by enterprises [4]. But few of the architectures exist only for the big data systems. In order to implement big data on cloud domain, researchers in [23] proposed a service model. The high level architecture has been proposed by Demchenko et al. for Big Data systems with high description for basic infrastructure [30]. The authors presented a framework which contains classification space that is multi-dimensional [1]. The design claims that architectures with multi-dimensional classification space leads to better results and success. A similar framework has been presented in [18] which provides an empirical design of a software system. The empirical data has been collected through different research methods viz. interviews, document analysis, questionnaires. This process is of linear type which flows through a definite step-wise manner starting with selection of architecture, use of design strategy, acquisition of related data empirically, carving out variability and at last the evaluation of system.

In order to optimize various growth patterns in big management systems Doshi et al. proposed an architecture by combining SQL and features of NewSQL [15]. The author in [19] designed a reference architecture and gave a detailed validation of the architecture by comparing the said architecture with Oracle, Facebook. The architecture was also empirically evaluated in reference to other already designed social networking architectures [18, 1].

One of high performance platform for streaming analysis is BlockMon that on basis of Call Data Records is analysed for call detection by telemarketers [9,16]. There are various number of designs available in current scenario which can be called as use cases of big data. The social networking sites viz LinkedIn, Facebook, Netflix and Twitter are the real time examples of such domains. The LinkedIn contains streamed and structured data. The data is fetched into the production and development based environments for analysis [24]. The data analytics model provides services like People You May Know which is also domain of data analytics [25]. The facebook follows batch analysis of streamed and structured data created by the people on this social networking site. For getting the deep inferences facebook scientists uses ad hoc analysis in developments environments and production systems [3]. The video streaming site Netflix collects user patterns and performs the analysis on user patterns in online or offline mode. The real time data analysis is performed which provides further video recommendation to the end user [28]. The traffic on network is also calculated using data analysis. The prime job of Twitter is to handle tweets and the incoming comments [3, 14, 11]. The “Who to Follow” service is also provided by the Twitter on basis of tweet and comment analysis [21]. Y. Lee provides details how to analyses and measure growing internet traffic [32]. The packet analysis for monitoring traffic on internet has been explored for better performance [33].

P.Paakonen et al. put forth a reference architecture which is technology independent for the systems that are using big data. The research work has been influenced by already use cases for big data systems [22]. Z. Ning *et al.* proposed an algorithm to schedule for scheduling spectrum and deep-reinforcement –learning based method [34].

In order to provide solution to the problems discussed above and objectives defined, a novel framework have been proposed in this research that is dealing with big data analytics related problems. The objective of the framework is to utilize Big Data as a service for big data mining related issues. This section provides a design and general architecture. Fig. 1 provides a detailed design of framework and its core parts.

III. General Architecture of the Framework

The framework proposed is technology independent. The algorithm has been implemented on Java based platform by using Netbeans. The input data files has been collected from Kaggle repository. The research generally provides a complete solution for how to deal with big data related issues in various technologies. But most significantly the proposed work will particularly handle the issues arising out of big data created on IoT (internet of Things). The techniques and methods in vogue created a bottleneck like situation for handling data coming from different sources. Therefore, keeping in view the drawbacks of current scenario, the framework have been proposed.

The general architecture of the proposed framework has been explained in Fig. 1. This is a technology independent framework for providing optimal solution to data driven applications. The growing use of data by organizations for making trend analysis is expanding rapidly. The demands of exponential use of various types of data has made the current technologies insufficient to deal with. The research proposed provides a generalized design to deal with variety of data created by online applications. The framework handles the data from its inception till a decision is made out of it. The prime aim of this framework is to provide a novel design for handling the data burst from multiple sources. As soon the input data is fetch into the architecture at very first instance the data is pre-processed to make it finished with respect to the system. The clustering of data is done in two phases. In first phase domain specific clusters are formed and in second phase the node specific clusters are formed inside a particular domain. The domain specific clusters are distributed into their respective domains by using Data Distribution Algorithm.

Fig. 1 depicts a detailed aspects of the data movement across the framework. When data is fetch from source viz IOT, Social Networking Site, etc. the same is transferred into the data preprocessing unit of the architecture. In the data preprocessing unit some primary data realizations are performed. The un-used instances in data are removed and data is set in a format to be used in hierarchy of the architecture. After cleaning of data for various types of redundancies and inaccuracies, the clusters are formed according to the domains available with the system. The data cluster is accordingly transferred into the respective domain using Domain Specific Data Distribution Algorithm.

Data Distribution in process as shown in Fig. 1 is a generalized process of distribution of clusters in their respective domains. The clusters formed are distributed into their respective domains using DSDDA (Domain Specific Data Distribution Algorithm). The clusters are having an index that matches with that of domain, which in turn helps in locating a domain specific cluster. The machine learning is process by which a system on basis of previous information forecasts the future decisions. The machine learning Module at this state plays the part by minimizing the overhead of indexing the cluster for which has already been done previously for a particular domain.

Fig. 2 represents the high level design of the proposed framework. The various modules working with the framework are defined. The section also provides the details about the how the designed system is evolved version of previous research. The domains are internally divided into nodes which represent a basic unit of domain. The basic tool is architecture which is made of data units and corresponding functions which access this data. These nodes store clusters of its specificity. The data stored in the nodes is accessed by the modules wrapped around the node.

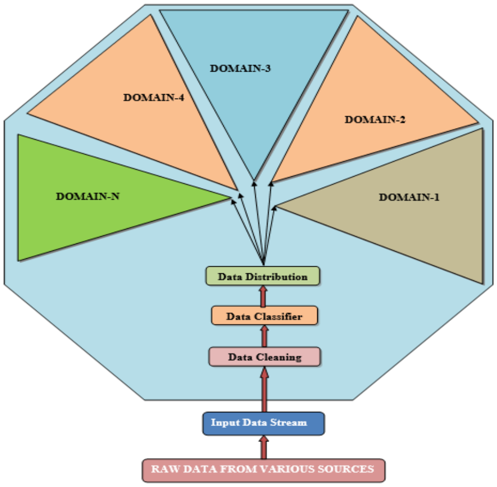

Fig. 1. DSBIGDA Framework.

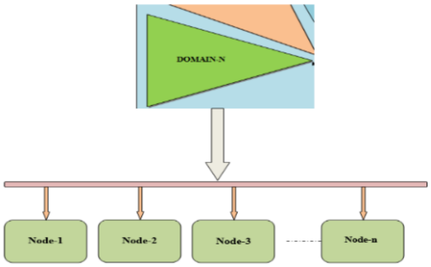

Fig. 2. Internal Structure of Domain.

A. Proposed Framework and Parallelization

Parallelization is important phenomenon in the proposed framework. In order to make the process of data distribution among various domains real time, it is necessary to provide a base for parallelism so that the incoming speed can be handled properly. In addition to parallelization module, machine learning module helps in reducing the time complexity for already fetched information.

B. Design of the Reference Architecture

Fig. 3 represents a general layout of domain specific data distribution. The rectangles represent the functionalities linked with the system and arrows are providing the actual flow of the data through various stages of the system. The data processing has been showed. The channel of transmitting the data during processing is starting from left and moves towards right direction. The data processing unit has been divided into individual areas with different functionalities. The scheduling of various processing units and their respective design have been kept separately in the framework. The streams of data provides information about the real time data generated by online social networking sites. The data of type structured nature has a dedicated data design model whereas, unstructured data has not any kind of design model associated with it i.e., relational databases. The content data from web pages is an example of unstructured type [13]. The data of type Semi-structured has flexible model with irregularities. XML documents is one example of this type of data [26].

The process of extraction is to get the data into the shape for inputting into the system. The data is stored temporarily and accordingly transferred as raw information into the store.

The process of compression is used to improve the efficiency while transferring the data during load operations. The raw information if any present with data during the load operations is cleaned for various variations and redundancies. This cleaned data is stored in separate file for making non-redundant input for next modules in the hierarchy. The main operations performed after clearing of input information is analytics for extracting some new information for decision making which is latter on stored in a structured format. The prime purpose of keeping different data stores at different modules is to hold the finished data and analysis which is achieved by executing the batch of jobs in a regular fashion.

The nodes internal to a particular domain performs the deep data analytics on stored information for reliable extraction. The copy of result is also stored into the basic data stores which latter on are used to publish the reports on the basis of result analysis. The data which is streamed in from online sources is used for general visualization. The supporting machine learning tools that are incorporated in the system are used to train the models for new patterns invoked with the data. The data from result analysis is further transformed for purpose of decision making in the relevant field.

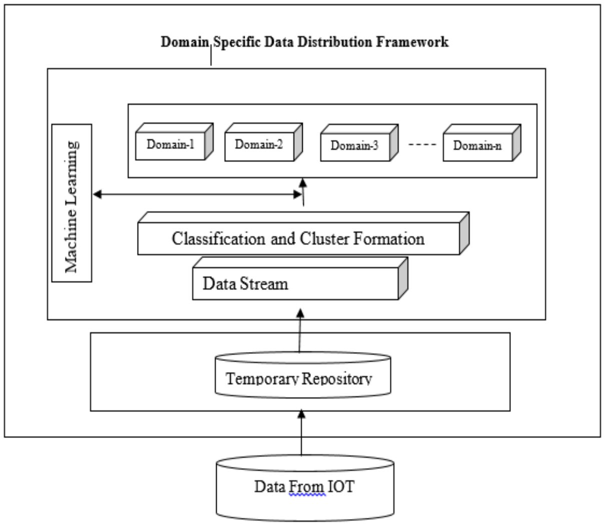

Fig. 3. General Layout of Domain Specific Data Distribution.

C. Domain Specific Data Distribution Algorithm

Algorithm DSDDA (*New, MAX, Dom[a], Dtype, Dvalue)*

Inputs:
New // Store value of an undefined domain
MAX // Total No of domains available
Dom[a] // Whether domain is available
DValue // the value of input data a
Dtype // the type of input data a
Outputs:
Result[a] // Total number of domain available with the framework after each new entry.

Main Procedure:

```
START of Data Distribution process, set a = 1, i= 1 /* executed on each distributed processor
The type and value of input data stream is copied into variable Dtype and DValue; // provides type-value pair of input data.
While index!=MAX // Loops through available domains
 if Dom[a] = Dtype
  Select the Dom[i] and store value of input data-value pair into the selected domain.
 End
index++;
a++;
End
if index= MAX // this condition is met only if the input data stream doesn't match with any already available domain
  MAX=MAX+1;
  Dom[MAX]=Dtype; // A new domain is created for the data that doesn't match
End
end distribution processing
 exit
```

The algorithm proposed is a generalized way of how the input data is distributed among the individual domains of the system. The data collector in the form of files is first fetch into the system to remove the redundancies in the data. The cleaned data is latter processed for making clusters in the form of data- value pair. These clusters are latter fetched into the information system as per the proposed algorithm.

The general flow of Domain Specific Data Distribution Algorithm is shown in Fig. 4. The program starts by taking input from source which can be real time data streams. In the next stage the data is preprocessed to make compatible to be taken as input and are latter fed to feed a domain and at same time the machine learning programs trains its modules for current data input into the system.

When the system finds a domain for its data the same is transferred to the respective domain. In the domain the data is fetched into the nodes as per the availability and node specification. In the nodes the data is actually used for analytics.

The data clusters are initially fetched into the machine learning module for making an intelligent inference to select a specific domain, where if the respective domain is not available the data is passed on to next steps of DSDDA Algorithm for creating a domain so that a cluster can be adjusted accordingly.

Each domain is internally divided into n-number of nodes. A node is a data analytical unit in which data is stored in the core and the analytical programs are surrounding it. The nodes provide a complete tool for storing, fetching and analyzing data of its related domain while storing data into it.

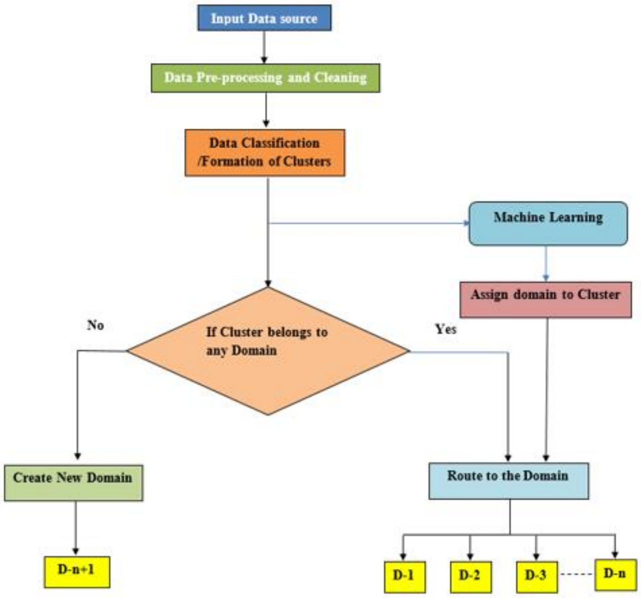

Fig. 4. Flow Chart of Domain Specific Data Distribution.

IV. Results and Discussion

This section provides a detailed comparison by considering the various performance metrics for evaluating the overall functionality of proposed system. The throughput as shown in Fig. 5 is comparatively far better than traditional frameworks.

The generalized function which is involved with data applications is getting input from a source and writing the results into the output system. The functionalities were discussed with some basic performance metrics measures as shown in Table I are compared. The input size has been kept in a range of 15.1x 10^5range.

The results shows that overall throughput of the proposed framework comparatively far better than the traditional framework. The input data was set at 1.5 * 10^5 and the number of operations per second is far better.

TABLE I. Performance Metrics of Proposed Framework

Variables Quantified	**Traditional RDBMS Based systems Sql Server**	**Domain Specific Big Data Framework**	**Traditional RDBMS Based systems**	**Domain Specific Big Data Framework**
Throughput(no. of Operations Per milliseconds)	5619	131075	5856	195321
Runtime(in milliseconds)	5500	410	7320	521
Average Latency(in milliseconds)	19.324	1.952	28.31	3.021
Min Latency(in milliseconds)	7.75	0.521	7.51	0.513
Max Latency(in milliseconds)	3250	172	5650	304
99th Percentile Latency(in milliseconds)	91.53	4.9201	118.32	5.928

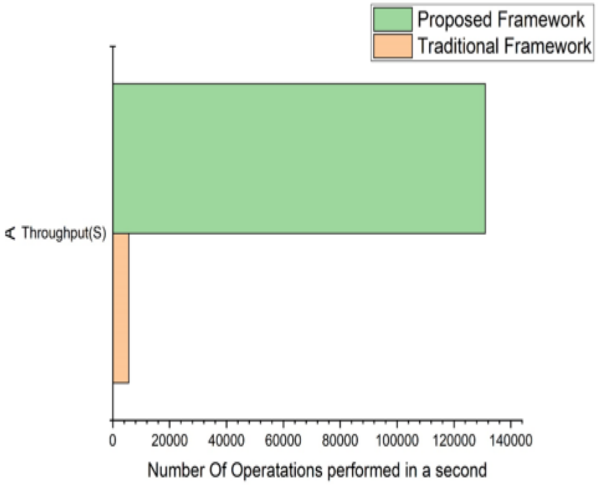

Fig. 5. Throughput of Proposed Framework.

Similarly for other metrics the results are shown in Fig. 6. The proposed scenario clearly shows that traditional approach is obsolete to deal with the growing needs of data. There should be a step in technology in order to meet the demands of exponential flow of information. This increase in use of data through enhanced technology will help in dealing with future decision making and predictions. The prior risk analysis is possible by using updated use of technology will provide the feasibility whether to carry on the project under consideration.

The output operations generalizations are showing in Fig. 7. The results completely show that the proposed framework performs efficiently than the traditional architectures so far as technology independent frameworks for handling big data is concerned.

In the premises the following results shown in Table II are inferred from the given results. The scalability of system is reliable which makes system fault tolerant with increase in number of nodes. The system is efficient while considering the operations on big data. The precision input/output is far better than the traditional mechanisms. Similarly the system error rate is comparatively negligible as compared to the existing systems.

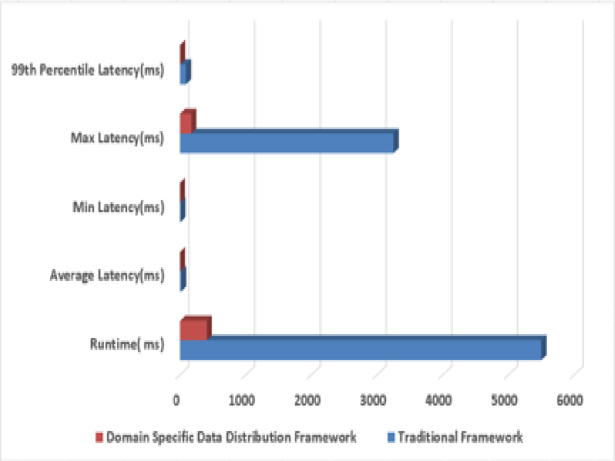

Fig. 6. Performance Metrics of Proposed System for Read Operations.

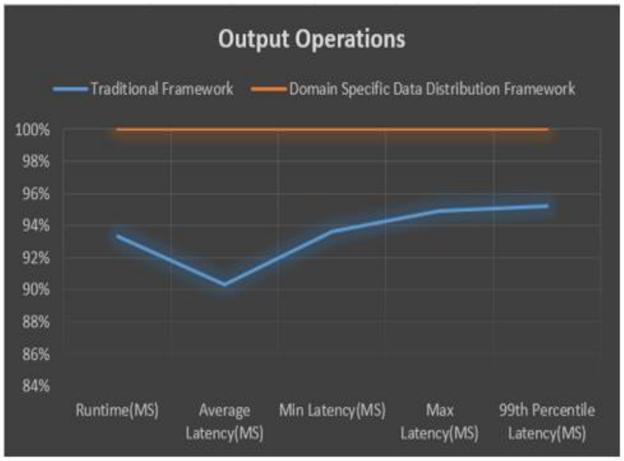

Fig. 7. Output Operations Performance Metrics of Proposed System.

TABLE II. OVERALL PERFORMANCE

System Used	**Scalability**	**Efficiency**	**Precision**	**Error Rate**	**Data Used**
Traditional Framework	Limited	Good	Less	>=10%	Big Data
Proposed Framework	Reliable	Excellent	More	<1%	Big Data

V. CONCLUSION AND FUTURE SCOPE

The framework put forth in this research is technology independent framework. The main aim of taking this research is to provide a basic design framework for solving the problems arising out of growing use of data science. The purpose of taking this research is to look into the growing issues with fast creation of data and non-availability of related technology. The prime issues that arise out of data is generally related to accuracy, efficiency and storage. These are issues have been addressed in the proposed framework by creating a platform that handles. The analytics of the stored data through machine learning process is adding a scope for dealing with AI (Artificial Intelligence) related issues. The design can be used to make a real time integrated system for dealing with data science analytics and prediction of future decisions arising out of huge data created on the internet.

REFERENCES

[1] Angelov, P. Grefen, and D. Greefhorst, “A framework for analysis and design of software reference architectures,” Inf. Softw. Technol. 54 (2012) 417–431.

[2] A. Cuzzocrea, D. Sacca, and J.D. Ullman, “Big data: a research agenda”, in: The 17th International Database Engineering & Applications Symposium, Barcelona, Spain, 09–11 October, 2013.

[3] A. Thusoo, et al., “Data warehousing and analytics infrastructure at Facebook,” in: 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, Indiana, USA, 6–11 June, 2010.

[4] A. Zimmermann, et al., “Towards and integrated service-oriented reference enterprise architecture,” in: International Workshop on Ecosystem Architectures, Saint Petersburg, Russia, 19 August, 2013.

[5] C.E. Cuesta, M.A. Martinez-Prieto, and J.D. Fernandez, “Towards an architecture for managing big semantic data in real-time,” in: 7th European Conference on Software Architecture, Montpellier, France, 1–5 July, 2013.

[6] C.L.P. Chen, and C. Zhang, “Data-intensive applications, challenges, techniques and technologies: a survey on Big Data,” Inf. Sci. 275 (2014) 314–347.

[7] C. Wang, K. Schwan, V. Talwar, G. Eisenhauer, L. Hu, and M. Wolf, "A Flexible Architecture Integrating Monitoring and Analytics for Managing Large-Scale Data Centers,” in the proceedings of ACM ICAC’11, 2011, Germany, pp. 141 -150.

[8] C. Zeng, et al., “FIU-miner: a fast, integrated, and user-friendly system for data mining in distributed environment,” in: 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, 11–14 August, 2013.

[9] D. Simoncelli, M. Dusi, F. Gringoli, and S. Niccolini, “Stream-monitoring with BlockMon: convergence of network measurements and data analytics platforms,” ACM SIGCOMM Commun. Rev. 43 (2013) 29–35.

[10] E. Begoli, “A short survey on the state of the art in architectures and platforms for large scale data analysis and knowledge discovery from data,” in: The 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA), Helsinki, Finland, 20–24 August,2012.

[11] G.L. Lee, J. Lin, C. Liu, A. Lorek, and D. Ryaboy, “The unified logging infrastructure for data analytics at Twitter,” in: The 38th International Conference on Very Large Databases, Istanbul, Turkey, 27–31 August, 2012.

[12] G. Mishne, “Fast data in the era of big data: Twitter’s real-time related query suggestion architecture,” in: The 2013 ACM SIGMOD International Conference on Management of Data, New York, New York, USA, 22–27 June, 2013.

[13] J. Chen, et al., “Big data challenge: a data management perspective,” Front. Comput. Sci. 2 (2013) 157–164.

[14] J. Lin, D. Ryaboy, “Scaling big data mining infrastructure: the Twitter experience,” ACM SIGKDD Explor. Newsl. 14 (2013) 6–19.

[15] K.A. Doshi, et al., “Blending SQL and NewSQL approaches reference architectures for enterprise big data challenges,” in: The International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, 10–12 October, 2013.

[16] M. Chen, S. Mao, and Y. Liu, “Big data: a survey,” Mob. Netw. Appl. 18 (2014).

[17] M. Dusi, et al., “BlockMon: flexible and high- performance big data stream analytics platform and its use cases,” NEC Tech. J. 7 (2012) 102–106.

[18] M. Galster, and P. Avgeriou, “Empirically-grounded reference architectures: a proposal,” in: Joint ACM SIGSOFT Conference on Quality of Software Architectures and ACM SIGSOFT Conference on Quality of Software Architectures and ACM SIGSOFT Symposium on Architecting Critical Systems, Boulder, Colorado, USA, June 20–24, 2011.

[19] M. Meier, “Towards a big data reference architecture,” Master’s thesis, Eindhoven University of Technology, October 2013.

[20] M. Westerlund, et al., “A generalized scalable software architecture for analyzing temporally structured big data in the cloud,” New Perspect. Inform. Syst. Technol. 1 (2014) 559–569.

[21] P. Gupta, et al., “WTF: the who to follow service at Twitter,” in: The International World Wide Web Conference, Rio de Janeiro, Brazil, 13–17 May, 2013.

[22] P.Paakonen, and D. Pakkala, “Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems,” Big Data Research 2 (2015) 166–186.

[23] Rabi et al., "Solving Big Data Challenges for Enterprise Application Performance Management," 38th Intl. Conference on Very Large Databases, pp. 1724-1735, 2012.

[24] R. Schmidt, M. Möhring, “Strategic alignment of cloud-based architectures for big data,” in: 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, Vancouver, Canada, 9–13 September, 2013.

[25] R. Sumbaly, J. Kreps, and S. Shah, “The “Big Data” Ecosystem at LinkedIn, in: 2013 ACM SIGMOD International Conference on Management of Data,” New York, New York, USA, 22–27 June, 2013.

[26] S. Abiteboul, “Querying semi-structured data,” in: International Conference onDatabase Theory, Delphi, Greece, 8–10 January, 1997.

[27] T. Zhong, et al., “On mixing high-speed updates and in-memory queries a bigdata architecture for real-time analytics,” in: IEEE International Conference on Big Data, Santa Clara, California, USA, 6–9 October, 2013.

[28] X. Amatriain, “Big & Personal: data and models behind Netflix recommendations,” in: The 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, Chicago, Illinois, USA, 11 August, 2013.

[29] X. Wu, G. Wu, W. Ding, “Data mining with big data,” IEEE Trans. Knowl. DataEng. 28 (2014) 97–106.

[30] Xiaoxing Gao, Hanxin Wang, and Xiaoxia Li "Application of Open-Source Big-Data Framework in Marine Information Processing," Journal of Coastal Research 98(sp1), 187-190, (27 December 2019). https://doi.org/10.2112/SI98-046.1

[31] Y. Demchenko, and C. Ngo, P. Membrey, “Architecture framework and components for the Big Data Ecosystem,” SNE Technical Report, University of Amsterdam, September 12, 2013.

[32] Y. Lee, and Y. Lee, “Toward scalable internet traffic measurement and analysis with hadoop,” ACM SIGCOMM Commun. Rev. 43 (2013) 5–13.

[33] Y. Lee, W. Kang, and Y. Lee, “A hadoop-based packet trace processing tool, in: International Workshop on Traffic Monitoring and Analysis,” Vienna, Austria, April 27, 2011.

[34] Z. Ning et al., "When Deep Reinforcement Learning Meets 5G-Enabled Vehicular Networks: A Distributed Offloading Framework for Traffic Big Data," in IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 1352-1361, Feb. 2020f