Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol.7 No.1, 2018, pp. 112-118

Importance of MapReduce for Big Data Applications: A Survey

M. Durairaj[1] and T. S. Poornappriya[2]
[1]Assistant Professor, [2]Research Scholar, [1&2]School of Computer Science, Engineering and Applications,
Bharathidasan University, Trichy, Tamil Nadu, India
E-Mail: poorna.priya23@gmail.com

***Abstract* - Significant regard for MapReduce framework has been trapped by a wide range of areas. It is presently a practical model for data-focused applications because of its basic interface of programming, high elasticity, and capacity to withstand the subjection to defects. Additionally, it is fit for preparing a high extent of data in Distributed Computing environments (DCE). MapReduce, on various events, has turned out to be material to a wide scope of areas. MapReduce is a parallel programming model and a related usage presented by Google. In the programming model, a client determines the calculation by two capacities, Map and Reduce. The basic MapReduce library consequently parallelizes the calculation and handles muddled issues like data dispersion, load adjusting, and adaptation to non-critical failure. Huge data spread crosswise over numerous machines, need to parallelize. Moves the data, and gives booking, adaptation to non-critical failure. A writing survey on the MapReduce programming in different areas has completed in this paper. An examination course has been distinguished by utilizing a writing audit.**
Keywords: **Big Data, Hadoop, Distributed File System, MapReduce Programming, Cloud Computing**

I. INTRODUCTION

These days, with the extreme development in the data and information, their investigation has turned into a troublesome test. MapReduce is blaming tolerant, basic, and adaptable structure for data handling that empowers its clients to process these enormous measures of data [1]. It is a system for effective expansive scale data handling which is exhibited by Google in 2004 so as to handle the issue of preparing a lot of data with reference to the Internet-based applications [2]. These extensive information data should be filed, put away, recovered, broke down and furthermore mined to permit a straightforward and proceeds with access to these data and data [3]. MapReduce is one of the precursors in the supposed "NoSQL" pattern to guide it far from standard social databases [4]. These days, there are four components, including preparing, putting away, perception, and investigating huge data in present-day associations and endeavors. The MapReduce can naturally run the applications on a parallel group of equipment and likewise, it can process terabytes and petabytes of data all the more quickly and effectively. Subsequently, its prominence has developed quickly for different brands of endeavors in numerous fields. It gives a profoundly powerful and effective structure for the parallel execution of the applications, data allotment in distributed database systems, and adaptation to internal failure arrange interchanges [5][6].

The principal goal of MapReduce is to encourage data parallelization, data appropriation and burden adjusting in a basic library [7]. The simple accessibility and availability of the MapReduce stages, for example, Hadoop, makes it adequate for a profitable parallelization and execution of data-concentrated undertakings [8]. The utilized algorithms in the MapReduce have a stipend to deal with the data-concentrated applications, parallel executions, and blame control instruments [9]. Software engineers who utilize the library of MapReduce must think about two capacities, for example, a Map and a Reduce work [10][11]. The Map work gets a key/esteem pair as information and makes the middle of the road key/esteem sets for additional preparing. Reduce work combines all the middle key/esteem matches and afterward makes the last yield [12].

II. BACKGROUND STUDY

A. Big Data

Over the late years, the volume and unpredictability of collaborations between data systems have been consistently expanding [4]. Big data examination (arranging, gathering and dissecting tremendous sets of data) are among the present most much of the time talking about themes in research and practice [13]. The world is being changed by data-driven methodologies including access to a lot of data and accessible open doors in business, science, and computing applications [14][15]. This class of utilization is significantly parallel and appropriate for the MapReduce programming that gives clients a chance to execute substantial scale data investigations with the end goal that the application execution layer handles the assignment planning, system engineering and data dividing [16]. The same number of ventures and associations handle expanding measures of data, big data preparing is being considered as the most essential advance. Right now, distributed computing systems are by and large generally utilized for big data handling. These systems enable the client to compose applications through a lot of abnormal state tasks and naturally handle the mind-boggling parts of distributed computing, for example, planning and adaptation to non-critical failure [17][18]. Apache's Hadoop and Google's MapReduce, its open-source usage, are the true programming systems for big-data applications. The MapReduce system expects to make a gigantic measure of the middle of the road data. Big data applications take a substantial amount of info data in the majority of the

applications [19]. Big data, for the most part, allude to a heterogeneous class of business applications that work on a lot of data [20]. It likewise has difficulties related three essential features: volume, speed, and assortment. Big Data has 3Vs; Volume (a lot of data), Velocity (data lands at fast) and Variety (heterogeneous assets). In big data definition, big alludes to a dataset which makes data idea of developing so much that it ends up hard to oversee it by utilizing existing data the board ideas and apparatuses. MapReduce is assuming a noteworthy job in the preparing of big data is an evident truth [21]. Since MapReduce gains a great deal of prevalence for its adaptation to non-critical failure, adaptability, versatility, and straightforwardness, it turns into the most appropriate structure for preparing an investigation of big data tasks.

B. Hadoop Distributed File System

Hadoop gives a distributed file system and a structure for the investigation and change of extremely expansive data sets utilizing the MapReduce worldview. An essential normal for Hadoop is the parceling of data and calculation crosswise over a large number of hosts and executing application calculations in parallel near their data. A Hadoop group scales calculation limit, stockpiling limit, and IO transfer speed by basically including item serversHadoop give a distributed file system and a structure for the examination and change of extremely expansive data sets utilizing the MapReduce worldview. A critical normal for Hadoop is the parceling of data and calculation crosswise over a huge number of hosts and executing application calculations in parallel near their data. A Hadoop bunch scales calculation limit, stockpiling limit, and IO data transfer capacity by basically including product servers.
HDFS is the file system part of Hadoop. While the interface to HDFS is designed after the UNIX file system, devotion to measures was yielded for improved execution for the current applications. HDFS stores file system metadata and application data independently.

C. MapReduce Architecture

Google made MapReduce process a lot of unstructured or semi-organized data, for example, web reports and logs of site page demands, on substantial shared-nothing bunches of ware hubs. It created different sorts of data, for example, altered records or URL get to frequencies [22]. The MapReduce has three noteworthy parts, including Master, Map capacity and Reduce work. The Master is in charge of dealing with the back-end Map and Reduce capacities and offering data and systems to them [23][24]. A MapReduce application contains a work process of employment where each activity makes two clients determined capacities: Map and Reduce. The Map work is connected to each info record and delivers a rundown of transitional records.

The Reduce work (likewise called Reducer) is connected to each gathering of the middle of the road records with a similar key and delivers a rundown of yield records [17]. MapReduce program is required to be done on a few PCs and hubs when it is performed on Hadoop [25]. Consequently, an ace hub runs all the important administrations to arrange the correspondence among Mappers and Reducers. An info file (or files) is isolated into similar parts called input parts. They go to the Mappers in which they work parallel together to give the data contained inside each split. As the data is given by the Mappers, they separate the yield; at that point, every Reducer assembles the data parcel by every Mapper, consolidates them, forms them, and produces the yield file.

The fundamental periods of MapReduce design are Mapper, Reducer, and mix which are introduced beneath:

1. Mapper: The Mapper forms to input data which are relegated by the ace to play out some calculation on this information and produce middle of the road results as key/esteem sets [26].
2. Reducer: The Reduce work gets a halfway key and a lot of estimations of the key. It joins these qualities together to shape a lesser arrangement of qualities [27].

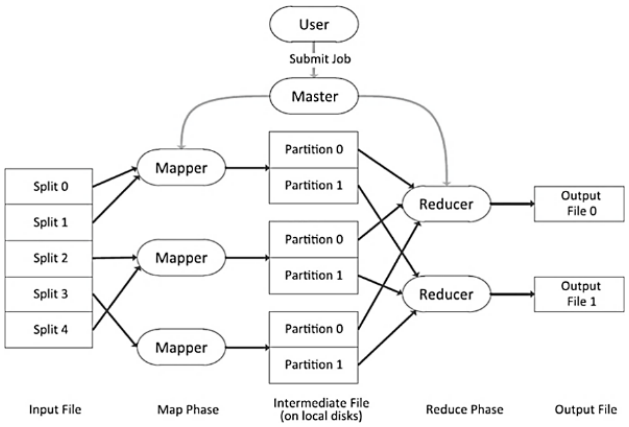

Fig. 1 Architecture of MapReduce Operation

III. LITERATURE SURVEY ON MAP REDUCE MODEL IN VARIOUS APPLICATIONS

Giachetta, Roberto [28] portrayed a geospatial data preparing structure intended to empower the administration and handling of spatial and remote detecting data in distributed condition. The system depends on the standard MapReduce worldview, and its open-source execution, the Apache Hadoop library.

Jin, Songchang, *et al.*, [29] build up a novel distributed network structure mining system. The writers proposed another connection data refresh strategy to endeavor to maintain a strategic distance from data composing related tasks and attempt to speed up the procedure. They utilized the nearby data from the hubs and their neighbors, rather than the page rank, to ascertain the likelihood dispersion of the hubs. It rejects the system parcel procedure and endeavors to run the guide condition specifically on MapReduce.

Zhang, Fan, *et al.,* [30] proposed an errand level versatile MapReduce structure. This structure expands the nonexclusive MapReduce engineering by planning each Map and Reduce task as a steady running circle daemon. The magnificence of this new structure is the scaling ability being planned at the Map and Task level, instead of being scaled from the register hub level.

Landset, Sara, *et al.,* [31] proposed to help the specialist or expert who comprehends AI yet is unpracticed with big data.

López, Victoria, *et al.,* [32] proposed the Chi-FRBCS-BigDataCS calculation, a fluffy principle based classification system that can manage the uncertainty that is an introduction in extensive volumes of data without ignoring the learning in the underrepresented class. The strategy utilizes the MapReduce system to circulate the computational activities of the fluffy model while it incorporates cost-delicate learning procedures in its plan to address the unevenness that is available in the data.

Mashayekhy, Lena, *et al.,* [33] proposed a structure for improving the vitality effectiveness of MapReduce applications while fulfilling the administration level understanding (SLA). The creator first model the issue of vitality mindful planning of a solitary MapReduce work as an Integer Program. The creator at that point proposes two heuristic algorithms, called Energy-mindful MapReduce Scheduling Algorithms (EMRSA-I and EMRSA-II), that discover the assignments of the guide and lessen undertakings to the machine spaces so as to limit the vitality expended when executing the application.

Peralta, Daniel, *et al.,* [34] showed a feature selection calculation dependent on the developmental calculation that utilizes the MapReduce worldview to acquire subsets of features from big datasets. The calculation decays the first dataset in squares of occurrences to gain from them in the guide stage; at that point, the decrease stage consolidates the acquired fractional outcomes into the last vector of feature loads, which permits an adaptable use of the feature selection system utilizing a limit to decide the chose subset of features.

Triguero, Isaac, *et al.,* [35] built up a MapReduce-based structure to circulate the working of these algorithms through a group of computing components, proposing a few algorithmic techniques to coordinate numerous fractional arrangements (decreased sets of models) into a solitary one.

Yao, Qin, *et al.,* [36] manufactured a five-hub Hadoop group to execute distributed MapReduce algorithms. The distributed algorithms show guarantee in encouraging effective data handling with therapeutic big data in social insurance administrations and clinical research contrasted and single hubs.

Wang, Yong, *et al.,* [37] proposed an uncommonly structured MapReduce calculation with lattice record to diminish the running time. The proposed calculation can diminish the seasons of calling convergence calculation by the guide of the matrix list.

Bechini, Alessio, Francesco Marcelloni, and Armando Segatori [38] proposed a distributed affiliation rule-based classification plot formed by the MapReduce programming model. The plan mines classification affiliation rules (CARs) utilizing a legitimately upgraded, distributed adaptation of the notable FP-Growth calculation.

Tsai, Chih-Fong, Wei-Chao Lin, and Shih-Wen Ke [39] expected to think about the execution contrasts between the distributed and MapReduce techniques over expansive scale datasets as far as mining exactness and proficiency. The examinations depend on four huge scale datasets, which are utilized for the data classification issues.

Cao, Jianfang, *et al.,* [40] proposed a parallel plan and acknowledgment technique for a molecule swarm improvement (PSO)- upgraded BP neural system dependent on MapReduce on the Hadoop stage utilizing both the PSO calculation and a parallel structure. The PSO calculation was utilized to advance the BP neural system's underlying loads and limits and improve the exactness of the classification calculation.

Kamal, Sarwar, *et al.,* [41] examined a contemporary distributed bunching technique for lopsidedness data decrease utilizing k-closest neighbor (K-NN) classification approach has been presented. The significant goal of this work is to show genuine preparing data sets with a decreased number of components or occurrences.

Gu, Boncheol, *et al.,* [42] presented Biscuit, a novel close data handling system intended for current strong state drives. It enables software engineers to compose a data-serious application to keep running on the host system and the capacity system in a distributed, yet consistent way.

Chen, Jiaoyan, *et al.,* [43] proposed a MapReduce-based distributed structure named MR-ELM to empower extensive scale Extreme Learning Machine (ELM) preparing. Under the system, ELM submodels are prepared parallelly with the distributed data obstructs on the group and afterward consolidated as a total single-shrouded layer feedforward neural system.

Xia, Yingjie, *et al.,* [44] proposed big traffic data handling structure utilizing HBase to investigate the data of Intelligent Monitoring and Recording System (IMRS).

Kumar, Ajay, *et al.,* [45] proposed in this examination utilizes a crossbreed way to deal with an arrangement with big data-set for more astute choices. The proposed MapReduce structure has been utilized for blame

identification by overseeing data irregularity issue properly and relating it to the association's benefits work.

Eldawy, Ahmed, Mohamed F. Mokbel, and Christopher Jonathan [46] presented Hadoop Viz; a Map Reduce based system for envisioning big spatial data. Hadoop Viz has three remarkable features that recognize it from different procedures.

Zhai, Junhai, Xizhao Wang, and Xiaohe Pang [47] proposed the calculation utilizes the Map of MapReduce to parcel the vast data sets into some little subsets and conveys them to various cloud computing hubs.

Manogaran, Gunasekaran, *et al.,* [48] utilized a Bayesian shrouded Markov display (HMM) with Gaussian Mixture (GM) Clustering way to deal with the model the DNA duplicate number change over the genome. The proposed Bayesian HMM with GM Clustering approach is contrasted and different existing methodologies, for example, Pruned Exact Linear Time strategy, parallel division technique, and portion neighborhood strategy.

Sadikin, Rifki, *et al.,* [49] executed Hadoop Map-Reduce system for handling Next-Generation Sequencing utilizing Hadoop-BAM library. The execution procedure a Binary Alignment Map (BAM) file which contains a reference arrangement and many adjusted/not-adjusted peruses by spitting the BAM file into Hadoop data squares.

Li, Zhenlong, *et al.,* [50] proposed a spatiotemporal ordering way to deal with proficiently oversee and process big atmosphere data with MapReduce in an exceedingly adaptable condition. Utilizing this methodology, big atmosphere data are specifically put away in a Hadoop Distributed File System in its unique, local file design.

Ahmad, Awais, *et al.,* [51] exhibited a system design that improves the working of conventional MapReduce by joining parallel handling calculation. In addition, the complete four-level design is additionally recommended that productively total the data, take out superfluous data, and break down the data by the proposed parallel handling calculation.

Fernández, Alberto, *et al.,* [52] went for breaking down the interrelation between the number of names of the fluffy factors and the shortage of the data because of the data testing in MapReduce.

Benmounah, Zakaria, Souham Meshoul, and Mohamed Batouche [53] portrayed an exceptionally adaptable Differential Evolution (DE) calculation dependent on the MapReduce programming model. The conventional utilization of DE to manage a grouping of substantial sets of data is so tedious that it isn't practical. Then again, map-decrease is a programming model developed of late to permit the plan of parallel and distributed methodologies.

Zhai, Junhai, Sufang Zhang, and Chenxi Wang [54] proposed a novel calculation, which comprises of four phases: (1) then again over-example p times between positive class occurrences and negative class occasions; (2) build l adjusted data subsets dependent on the created positive class cases; (3) train l segment classifiers with outrageous learning machine (ELM) calculation on the built l adjusted data subsets; (4) coordinate the l ELM classifiers with basic casting a ballot approach.

Pulgar-Rubio, F., *et al.,* [55] exhibited another calculation for the subgroup revelation task called MEFASD-BD. The calculation is created in Apache Spark dependent on the MapReduce worldview, and it can handle high dimensional datasets in a productive way.

Cho, Wonhee, and Eunmi Choi [56] built up a MapReduce program utilizing a Hadoop ecosystem and conveyed big data to play out the pre-preparing investigation.

Zhang, Fan, *et al.,* [57] proposed an assignment level versatile MapReduce system. This system broadens the nonexclusive MapReduce engineering by structuring each Map and Reduce task as a versatile daemon process.

Talan, Pooja P., *et al.,* [58] proposed an adaptable chart preparing engineering that could be utilized to conquer the customary constraints of the Hadoop System and to give a similar examination of Hadoop MapReduce and Apache Spark.

Zhang, Bin, Xiaoyang Wang, and Zhigao Zheng [59] presented an improvement systems for repeating inquiries for big data investigation. Right off the bat, it examines the effect of repeating questions effectiveness by MapReduce repeating inquiries display. Furthermore, it proposes the MapReduce steady window cut calculation, which can not just make more open doors for reuse of repeating inquiries, yet in addition extraordinarily lessen repetitive data while stacking input data by the fine-grained booking. Thirdly, regarding data planning, it structures the MapReduce late booking methodology that improves data handling and enhances calculation asset planning in MapReduce group. At long last, it develops the effective data reuse execution designs by MapReduce repeating questions reuse methodology.

Qian, Jin, Min Xia, and Xiaodong Yue [60] proposed the quantitative measure changes of the help, certainty, and inclusion related with progressive choice tenets are additionally talked about to clarify these connections between the condition granules and choice granule. A productive parallel learning obtaining system utilizing MapReduce for big data is proposed and actualized.

Martín, D., *et al.,* [61] proposed MRQAR, another nonexclusive parallel structure to find quantitative affiliation leads in a lot of data, planned after the MapReduce worldview utilizing Apache Spark.

Manogaran, Gunasekaran, and Daphne Lopez [62] displayed a versatile data handling system with a novel change location calculation. The vast volume of atmosphere data is put away on Hadoop Distributed File System (HDFS) and MapReduce calculation is connected to ascertain the occasional normal of atmosphere parameters. Spatial autocorrelation-based environmental change location calculation is proposed in this paper to screen the adjustments in the regular atmosphere.

Zou, Quan, Guoqing Li, and Wenyang Yu [63] proposed a unique data configuration to accomplish the bound together portrayal of remote detecting data. The data reflection is to discretize the multidimensional remote detecting data for simple distributed capacity and calculation. Utilizing MapReduce worldview, the intricacy of remote detecting algorithms is settled.

Tran, Xuan T., *et al.,* [64] proposed another data design conspire that can be executed for HDFS. A correlation between our proposition and the current HDFS data format conspire demonstrates that the new data design calculation essentially decreases the vitality utilization at the slight cost of the mean reaction time of occupations.

Ramírez-Gallego, Sergio, *et al.,* [65] proposed a distributed discretization calculation for Big Data examination dependent on developmental streamlining.

Manogaran, Gunasekaran, Daphne Lopez, and Naveen Chilamkurti [66] proposed the big data handling structure to incorporate atmosphere and wellbeing data and to discover the relationship between's the atmosphere parameters and occurrence of dengue. This structure is exhibited with the assistance of MapReduce programming model, Hive, HBase and ArcGIS in a Hadoop Distributed File System (HDFS) condition.

Zhang, Liang, *et al.,* [67] presented a MapReduce structure, named HadInc, for effective gradual calculations. HadInc is intended for disconnected scenes, in which constant is unnecessary and in-memory bunch computing is invalid.

IV. FUTURE RESEARCH DIRECTION

As per this examination, the exploration bearing of MapReduce can be isolated into two headings, first course concerns upgrading MapReduce programming model, for instance, current MapReduce methods for dealing with skew of data still need more examination so as to handle key gathering load unevenness, where most of the data are doled out to few keys. Another upgrade to MapReduce is following hubs measurements to find explicit hubs that defer the execution time of employment. This deferral might be brought about by burden unevenness or equipment shortcomings. The second research bearing worries by changing over existing algorithms of various areas to keep running in MapReduce. This should be possible by software engineers that are limited to express the calculation in the guide and decrease works so the future research plans to change the momentum algorithms to be appropriate for MapReduce.

V. CONCLUSION

MapReduce has been imagined by Google to manage the gigantic volume of data. In this paper, we presented an outline of the MapReduce programming model. Utilization of MapReduce display in different areas, businesses are exhibited by this paper. Through this review paper, another examination pattern is building up a layer on MapReduce that convert current algorithms naturally or semi consequently to be reasonable to MapReduce programming model. Additionally, scientists can explore new algorithms that mull over MapReduce confinements.

REFERENCES

[1] Wang, Botao, *et al.,* "Parallel online sequential extreme learning machine based on MapReduce", *Neurocomputing,* Vol. 149, pp. 224-232, 2015.

[2] Marozzo, Fabrizio, Domenico Talia, and Paolo Trunfio. "P2P-MapReduce: Parallel data processing in dynamic Cloud environments." *Journal of Computer and System Sciences,* Vol. 78, No.5, pp. 1382-1402, 2012.

[3] Mohamed, Hisham, and Stéphane Marchand-Maillet. 'MRO-MPI: MapReduce overlapping using MPI and an optimized data exchange policy", *Parallel Computing,* Vol.39, No.12, pp. 851-866, 2013.

[4] Barre, Benjamin, *et al.,* "MapReduce for parallel trace validation of LTL properties", *International Conference on Runtime Verification*. Springer, Berlin, Heidelberg, 2012.

[5] Lu, Lu, *et al.,* "Morpho: A decoupled MapReduce framework for elastic cloud computing", *Future Generation Computer Systems,* Vol. 36, pp. 80-90, 2014.

[6] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: a flexible data processing tool", *Communications of the ACM,* Vol.53, No.1, pp. 72-77, 2010.

[7] Dean, Jeffrey, and Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters", *Communications of the ACM,* Vol. 51, No.1, pp. 107-113, 2008.

[8] Kolb, Lars, Andreas Thor, and Erhard Rahm, "Multi-pass sorted neighborhood blocking with MapReduce", *Computer Science-Research and Development,* Vol. 27, No.1, pp. 45-63, 2012.

[9] [9] Anjos, Julio CS, *et al.,* "MRA++: Scheduling and data placement on MapReduce for heterogeneous environments", *Future Generation Computer Systems,* Vol. 42, pp. 22-35, 2015.

[10] Zhang, Junbo, *et al.,* "A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems", *International Journal of Approximate Reasoning,* Vol.55 No.3, pp. 896-907, 2014.

[11] Slagter, Kenn, *et al.,* "SmartJoin: a network-aware multiway join for MapReduce", *Cluster Computing,* Vol. 17, No.3, pp. 629-641, 2014.

[12] Xiao, Zhifeng, and Yang Xiao, "Achieving accountable MapReduce in cloud computing", *Future Generation Computer Systems,* 30, pp.1-13, 2014.

[13] Debortoli, Stefan, Oliver Müller, and Jan vom Brocke, "Comparing business intelligence and big data skills", *Business & Information Systems Engineering,* Vol. 6, No.5, pp. 289-300, 2014.

[14] Shamsi, Jawwad, Muhammad Ali Khojaye, and Mohammad Ali Qasmi, "Data-intensive cloud computing: requirements, expectations, challenges, and solutions", *Journal of grid computing,* Vol.11, No.2, pp. 281-310, 2013.

[15] Lin, Jimmy, and Chris Dyer, "Data-intensive text processing with MapReduce", *Synthesis Lectures on Human Language Technologies,* Vol. 3, No.1, pp.1-177, 2010.

[16] Jain, Reshu, Prasenjit Sarkar, and Dinesh Subhraveti, "Gpfs-snc: An enterprise cluster file system for big data", *IBM Journal of Research and Development,* Vol. 57, No.3/4, pp. 5-1, 2013.

[17] Lee, Daewoo, Jin-Soo Kim, and Seungryoul Maeng, "Large-scale incremental processing with MapReduce", *Future Generation Computer Systems*, Vol. 36, 66-79, 2014.

[18] Zaharia, Matei, *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing", *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012.

[19] Zhao, Yaxiong, Jie Wu, and Cong Liu, "Dache: A data aware caching for big-data applications using the MapReduce framework", *Tsinghua science and technology*, Vol. 19, NO.1, pp. 39-50, 2014.

[20] Costa, Paolo, Austin Donnelly, Antony Rowstron, and Greg O'Shea, "Camdoop: Exploiting in-network aggregation for big data applications", In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pp. 29-42. 2012.

[21] Pandey, Shweta, and Vrinda Tokekar, "Prominence of MapReduce in big data processing", In *2014 Fourth IEEE International Conference on Communication Systems and Network Technologies*, pp. 555-560. IEEE, 2014.

[22] Liu, Ji, *et al.,* "A survey of data-intensive scientific workflow management", *Journal of Grid Computing*, Vol. 13, No.4, pp. 457-493, 2015.

[23] Wu, Tin-Yu, *et al.,* "Cloud-based image processing system with priority-based data distribution mechanism", *Computer Communications*, Vol. 35, No. 15, pp. 1809-1818, 2012.

[24] Senger, Hermes, *et al.,* "BSP cost and scalability analysis for MapReduce operations", *Concurrency and Computation: Practice and Experience*, Vol. 28,No .8, pp. 2503-2527, 2016.

[25] Idris, Muhammad, *et al.,* "Context-aware scheduling in MapReduce: a compact review", *Concurrency and Computation: Practice and Experience*, Vol. 27, No. 17, pp. 5332-5349, 2017.

[26] Lee, Chia-Wei, *et al.,* "A dynamic data placement strategy for hadoop in heterogeneous environments", *Big Data Research*, Vol. 1, pp. 14-22, 2014.

[27] Aridhi, Sabeur, *et al.,* "Density-based data partitioning strategy to approximate large-scale subgraph mining", *Information Systems*, Vol. 48, pp. 213-223, 2015.

[28] Giachetta, Roberto, "A framework for processing large scale geospatial and remote sensing data in MapReduce environment", *Computers & Graphics*, Vol. 49, pp. 37-46, 2015.

[29] Jin, Songchang, *et al.,* "Community structure mining in big data social media networks with MapReduce", *Cluster computing*, Vol. 18, No.3, pp. 999-1010, 2015.

[30] Zhang, Fan, *et al.,* "A task-level adaptive MapReduce framework for real-time streaming data in healthcare applications", *Future Generation Computer Systems*, Vol. 43, pp. 149-160, 2015.

[31] Landset, Sara, *et al.,* "A survey of open source tools for machine learning with big data in the Hadoop ecosystem", *Journal of Big Data*, Vol. 2, No.1, pp. 24, 2015.

[32] López, Victoria, et , "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data", *Fuzzy Sets and Systems*, Vol. 258, pp. 5-38, 2015.

[33] Mashayekhy, Lena, *et al.,* "Energy-aware scheduling of mapreduce jobs for big data applications", *IEEE transactions on Parallel and distributed systems*, Vol. 26, No.10, pp. 2720-2733, 2015.

[34] Peralta, Daniel, *et al.,* "Evolutionary feature selection for big data classification: A MapReduce approach", *Mathematical Problems in Engineering*, 2015.

[35] Triguero, Isaac, *et al.,* "MRPR: A MapReduce solution for prototype reduction in big data classification", *neurocomputing*, Vol. 150, pp. 331-345, 2015.

[36] [36] Yao, Qin, *et al.,* "Design and development of a medical big data processing system based on Hadoop", *Journal of medical systems*, Vol. 39, No.3, pp. 23, 2015.

[37] Wang, Yong, *et al.,* "Improving the performance of GIS polygon overlay computation with MapReduce for spatial big data processing", *Cluster Computing*, Vol. 18, No.2, 507-516, 2015.

[38] Bechini, Alessio, Francesco Marcelloni, and Armando Segatori, "A MapReduce solution for associative classification of big data", *Information Sciences*, Vol. 332, pp. 33-55, 2016.

[39] Tsai, Chih-Fong, Wei-Chao Lin, and Shih-Wen Ke, "Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies", *Journal of Systems and Software*, Vol. 122, pp. 83-92, 2016.

[40] Cao, Jianfang, *et al.,* "Big data: A parallel particle swarm optimization-back-propagation neural network algorithm based on MapReduce", *PloS one*, Vol. 11, No. 6, pp. e0157551, 2016.

[41] Kamal, Sarwar, *et al.,* "A MapReduce approach to diminish imbalance parameters for big deoxyribonucleic acid dataset", *Computer methods and programs in biomedicine*, Vol. 131, pp. 191-206, 2016.

[42] Gu, Boncheol, *et al.,* "Biscuit: A framework for near-data processing of big data workloads", *ACM SIGARCH Computer Architecture News IEEE Press*, Vol. 44. No. 3, 2016.

[43] Chen, Jiaoyan, *et al.,* "MR-ELM: a MapReduce-based framework for large-scale ELM training in big data era", *Neural Computing and Applications*, Vol. 27, No. 1, pp.101-110, 2016.

[44] Xia, Yingjie, *et al.,* "Big traffic data processing framework for intelligent monitoring and recording systems", *Neurocomputing*, Vol. 181, pp. 139-146, 2016.

[45] Kumar, Ajay, *et al.,* "A big data MapReduce framework for fault diagnosis in cloud-based manufacturing", *International Journal of Production Research*, Vol. 54, No. 23, pp. 7060-7073, 2016.

[46] Eldawy, Ahmed, Mohamed F. Mokbel, and Christopher Jonathan, "HadoopViz: A MapReduce framework for extensible visualization of big spatial data", *2016 IEEE 32nd International Conference on Data Engineering (ICDE), IEEE*, 2016.

[47] Zhai, Junhai, Xizhao Wang, and Xiaohe Pang, "Voting-based instance selection from large data sets with MapReduce and random weight networks", *Information Sciences*, Vol. 367, pp. 1066-1077, 2016.

[48] Manogaran, Gunasekaran, *et al.,* "Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering", *Wireless personal communications*, Vol. 102, No.3, pp. 2099-2116, 2018.

[49] Sadikin, Rifki, *et al.,* "Processing next generation sequencing data in map-reduce framework using hadoop-BAM in a computer cluster", *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 2017.

[50] Li, Zhenlong, *et al.,* "A spatiotemporal indexing approach for efficient processing of big array-based climate data with MapReduce", *International Journal of Geographical Information Science*, Vol. 31, No.1, pp. 17-35, 2017.

[51] [51] Ahmad, Awais, *et al.,* "Multilevel data processing using parallel algorithms for analyzing big data in high-performance computing", *International Journal of Parallel Programming*, pp. 1-20, 2018.

[52] Fernández, Alberto, *et al.,* "Fuzzy rule based classification systems for big data with MapReduce: granularity analysis", *Advances in Data Analysis and Classification*, Vol. 11, No.4, 711-730, 2017.

[53] Benmounah, Zakaria, Souham Meshoul, and Mohamed Batouche, "Scalable Differential Evolutionary Clustering Algorithm for Big Data Using Map-Reduce Paradigm", *International Journal of Applied Metaheuristic Computing (IJAMC)*, Vol. 8, No.1, pp. 45-60, 2017.

[54] Zhai, Junhai, Sufang Zhang, and Chenxi Wang, "The classification of imbalanced large data sets based on mapreduce and ensemble of elm classifiers", *International Journal of Machine Learning and Cybernetics*, Vol.8, No.3, pp. 1009-1017, 2017.

[55] Pulgar-Rubio, F., *et al.,* "MEFASD-BD: a multi-objective evolutionary fuzzy algorithm for subgroup discovery in big data environments-a MapReduce solution", *Knowledge-Based Systems*, Vol. 117, pp. 70-78, 2017.

[56] Cho, Wonhee, and Eunmi Choi, "Big data pre-processing methods with vehicle driving data using MapReduce techniques", *The Journal of Supercomputing*, Vol. 73, No.7, pp. 3179-3195, 2017.

[57] Zhang, Fan, *et al.,* "Process Streaming Healthcare Data with Adaptive MapReduce Framework", *Handbook of Large-Scale Distributed Computing in Smart Healthcare*. Springer, Cham, pp. 43-66, 2017.

[58] Talan, Pooja P., *et al.,* "An Overview of Hadoop MapReduce, Spark, and Scalable Graph Processing Architecture", *Recent Developments in Machine Learning and Data Analytics*. Springer, Singapore, pp. 35-42, 2019.

[59] Zhang, Bin, Xiaoyang Wang, and Zhigao Zheng, "The optimization for recurring queries in big data analysis system with

MapReduce", *Future Generation Computer Systems,* Vol. 87, pp. 549-556, 2018.

[60] Qian, Jin, Min Xia, and Xiaodong Yue, "Parallel knowledge acquisition algorithms for big data using MapReduce", *International Journal of Machine Learning and Cybernetics,* Vol. 9, No.6, pp. 1007-1021, 2018.

[61] Martín, D., *et al.*, "MRQAR: A generic MapReduce framework to discover quantitative association rules in big data problems", *Knowledge-Based Systems,* Vol. 153, pp. 176-192, 2018.

[62] Manogaran, Gunasekaran, and Daphne Lopez, "Spatial cumulative sum algorithm with big data analytics for climate change detection", *Computers & Electrical Engineering,* Vol. 65, pp. 207-221, 2018.

[63] Zou, Quan, Guoqing Li, and Wenyang Yu, "MapReduce functions to remote sensing distributed data processing—Global vegetation drought monitoring as an example", *Software: Practice and Experience*, Vol. 48, No.7, pp. 1352-1367, 2018.

[64] Tran, Xuan T., *et al.*, "A New Data Layout Scheme for Energy-Efficient MapReduce Processing Tasks", *Journal of Grid Computing*, Vol. 16, No.2, pp. 285-298, 2018.

[65] Ramírez-Gallego, Sergio, *et al.*, "A distributed evolutionary multivariate discretizer for big data processing on apache spark", *Swarm and Evolutionary Computation,* Vol. 38, pp. 240-250, 2018.

[66] Manogaran, Gunasekaran, Daphne Lopez, and Naveen Chilamkurti, "In-Mapper combiner based MapReduce algorithm for processing of big climate data", *Future Generation Computer Systems,* Vol. 86, pp. 433-445, 2018.

[67] Zhang, Liang, *et al.*, "Efficient finer-grained incremental processing with MapReduce for big data", *Future Generation Computer Systems,* Vol. 80, pp. 102-111, 2018.