International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2022 | Issue 52 | Volume 14 | Number 3 | Pages 212-220

A SPARK-BASED PARALLEL FUZZY C MEDIAN ALGORITHM FOR WEB LOG BIG DATA

M.A. Mallik [1] N.F. Zulkurnain [2] M.K. Nizamuddin [3] R. Sarkar [4] A.K. Chalil [5]

1. International Islamic University Malaysia, Kuala Lumpur, Malaysia and VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India, malamdot@gmail.com
2. International Islamic University Malaysia, Kuala Lumpur, Malaysia, nurulfariza@iium.edu.my
3. Shadan Women's College of Engineering and Technology, Jawaharlal Nehru Technological University, Hyderabad, India, mknizams@yahoo.com
4. University of Science and Technology, Meghalaya, India, sarkarrashel@gmail.com
[5] Malla Reddy College of Enginnering, Hyderabad,India,abswalih@gmail.com

Abstract- Now-a-days, the World Wide Web (WWW) is regarded as an exceptionally large data storehouse. The WWW is becoming more complicated and substantive every day. At the moment, the situation is such that we are starved for knowledge while drowning in data. Due to these factors, the data mining clustering technique is one of the most crucial tools for collecting useful data from the web. Clustering techniques for small datasets have led to the development of numerous successful clustering techniques. Nevertheless, these techniques do not provide adequate results when trading with extensive data sets. The most important problems are excessive computational difficulty and lengthy evaluating time, which is not acceptable for real-time context. It is very prime to process this enormous information on time. This paper proposes an efficient parallel Fuzzy C median solution based on Spark for large-scale web log data. Based on the Rand Index and SSE (sum of squared error), the parallel Fuzzy C median algorithm's performance is evaluated in the PySpark platform. According to the experimental findings, the parallel Fuzzy C median method built on Spark performs better.

Keywords: Fuzzy Clustering, Web Log Big Data, Parallel Computing, Apache Spark.

1. INTRODUCTION

Data mining's clustering technique involves categorizing data into numerous categories in order to obtain the necessary knowledge from a dataset [1]. There are two types of fuzzy system representation that have been created theoretically and are currently in practice. The first type is depicted as a fuzzy mathematical model, in which the state parameter clustering specifies the uncertainty.

The adoption of the second form of the fuzzy model makes it possible to support the fuzzy decisions. It is based on a series of fuzzy rules [2]. Fuzzy membership is promoted by fuzzy clustering. In fuzzy clustering, a single data set can belong to many clusters. It shows that a same data set can be used simultaneously in several clusters. Each data set's percentage of members in each cluster will be different; for instance, one data set may have a high membership while another data set may have a low membership. The membership value might be between 0 and 1. Each cluster center's session membership will be assumed to be one. Controlling data fuzzy clustering will help it more closely reflect reality. For instance, fuzzy clustering will assign partial cluster membership to a dataset if it lies on the border of two or more groups [3].

Nowadays, big data is very popular in the market. Big data mentions to extremely large-scale datasets especially brought together from distinct areas and pursue to extend at a high-speed pace [4]. Big Data is a vast body of information that has grown significantly over time. This data set is too big and complicated to store or analyze using conventional data management methods. Big data is far larger than regular data, although the two types of data are comparable. Digitized information is easy to capture and storing as it is very cheap the data storage frequency is developing at an exceptional rate. This developing data is amassed in various huge data storages. This sort of circumstance requires intense apparatuses to grasp knowledge from this ocean of information. With the very high growth of information sources accessible on the World Wide Web, it has ended up progressively vital for users to utilize programmed tools in locating the wanted data assets, and to track and analyze their usage patterns.

So, there is a necessity to create server-side and client-side tools that mine knowledge adequately, Cooley, et al. [5]. The study of user access patterns from web servers is known as web usage mining. How users are accessing a site is important to increase the utilization of the website by users. Preprocessing, pattern extraction, and outcome analysis are its three processes.

In the preprocessing stage, different types of noises are removed. The user and session identification process will be completed in this stage. A wide variety of pattern extraction techniques are available like clustering, path analysis etc. based on the needs of the analyst. Once web usage patterns are discovered there are different types of techniques and tools to analyze and understand them. Input web access logs contain a significant amount of unrelated data. The large number of user sessions and URL resources makes the dimension of web user session data very high. Human interactions and nondeterministic browsing patterns increase ambiguity and vagueness of user session data. The World Wide Web is a vast, dynamic information source with an incredibly sophisticated structural design that is always changing. Consequently, it is a useful setting for data or web mining. By using various data mining techniques, web mining can be used to obtain important knowledge from the internet. Web data frequently has unlabeled, dispersed, heterogeneous, semi-organized, time-shifting, and multi-dimensional features.

There are two kinds of algorithm of the clustering technique: Sequential computing algorithm and parallel computing algorithm. Sequential computing alludes to the program being practiced successively on a solitary processor. The simultaneous execution of a programmer on several processors is referred to as parallel computing. Most of the clustering methods suitable for parallel computing (Figure 1) are summarized in this document, which also offers current examination scenarios for the algorithms. The spirit of parallel computing is to gap and vanquishes enormous information. This paper generally reviews the clustering technique dependent on Spark processing system.

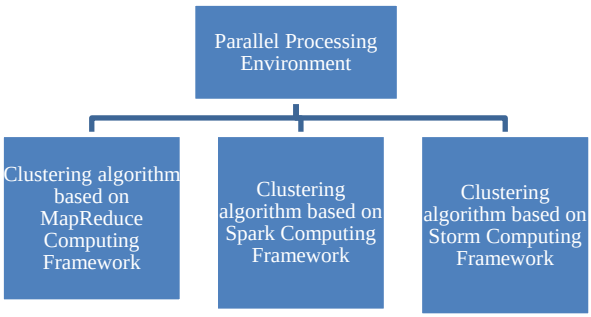

Figure 1. Categories of parallel processing environment [6]

Apache Spark is a distributed in-memory platform for iterative computation that caches frequently used data and preliminary findings in distributed memory. For some iterative algorithms, this greatly accelerates speed [7]. Apache Spark is an open-source platform for real-time processing cluster computing that was developed by the Apache Software Foundation. For the entire cluster, it provides a programming interface with implicit data parallelism and fault tolerance. It expands the MapReduce model to effectively utilize more categories of computations and was created on top of Hadoop MapReduce. The Apache Spark platform is used to implement the parallel fuzzy median clustering algorithm.

The suggested technique can completely use the FCM algorithm's intrinsic parallelism and speed up the segmentation speed of web log huge data, according to experimental results.

The Literature Review is presented in Area 2, the Framework and Algorithm is presented in Area 3, the Results are explained in Area 4, and the paper's conclusion is presented in Area 5.

2. LITERATURE REVIEW

2.1. Review of Parallel Clustering Algorithm

The performance of map and reduce for huge datasets was negatively impacted by the suggested K-Means Hadoop MapReduce (KM-HMR) clustering algorithm inability to provide huge datasets suitable work scheduling [8]. The suggested K-means clustering algorithm's distance computation function is a tedious and complex process [9]. The Apache Spark framework's K-means clustering method was developed, but it neglected to account for some important big data features, such as truthfulness and velocity, and it also failed to successfully cluster real-time streaming large data [10].

The modelled Accelerated MapReduce-based K-Prototypes (AMRKP) was unable to lower the iteration count or increase scalability in the used variation of the K-means clustering algorithm [11]. The Parallel K-Medoids Algorithm was developed to cluster vast data, but it did not function for clusters of large-scale data-intensive applications [12]. Due to the lack of parallelization in the initialization step while constructing a set of cluster centers in the newly proposed MapReduce-based K-Prototypes (MR-KP) clustering algorithm, it is inappropriate for real-time applications like fraud detection [13].

For handling gigabyte-sized data sets, the MapReduce fuzzy C-mean (MR-FCM) clustering technique was ineffective [14]. The designed Fuzzy c-means clustering algorithm must take effective action on how to extract common item sets because this is an important phase in data analysis. [15]. The weighted kernel Possibilistic c-Means (wkPCM) technique that has been developed for clustering vast volumes of data does not examine multi-dimensional space or deep features [16].

The High-order Possibilistic C-Mean (HOPCM) approach's effectiveness for highly scalable big data clustering can be improved [17]. The Clustering-based Collaborative Filtering (ClubCF) method did not overcome the scarcity issues, which might be made worse by semantic analysis for better coverage [18]. The use of the Apache Mahout may have improved the effectiveness of the recommendation generating process. For big data clusters, collaborative filtering technique still has room for improvement [19]. Due to its optimization-based character, the technique that used a genetic algorithm for big data clustering can still improve the efficacy of big data analytics [20]. The proposed hybrid evolutionary clustering strategy has a runtime penalty in comparison to previous methods [21]. The developed parallel ant colony clustering approach's significant iteration cost has an impact on the technique's performance [22].

2.2. Review of Web Usage Mining Using Fuzzy C Means Clustering

Web Usage Mining (WUM) is the study of how user interactions with a web server, such as web logs, click streams, and database transactions, affect a given website or a set of comparable websites. Preprocessing, pattern extraction, and results analysis are the three essential procedures carried out by WUM [24, 28, 29]. Giovanna, et al. [25, 26] preprocess web log data using a LODAP (Log Data Preprocessor) tool. For the purpose of removing superfluous log entries, identifying user accesses, and classifying user visits into user sessions, we analyze Web log data using LODAP, a piece of software.

The access data for the pages a user has accessed during a user session (number of accesses, time of visit, etc.) describes the user's navigational behavior. User identification is the process of identifying distinct users from online log information. Typically, the Extended Common Log file only contains the user agent and the IP address of the PC. Websites that demand user registration will include extra user login details that can be used to identify users. Each IP address will be regarded as a user if the user login information is not supplied. The next step is to identify user sessions. In this section, we will divide the web log data file into various segments known as user sessions. Every session is handled separately from other website visits. It is challenging to distinguish between user sessions in the web log file. These data can be used as input for numerous data mining algorithms [24, 28, 29].

The Fuzzy c-Means clustering algorithm is being used in this instance to group user sessions. Here we need to randomly select initial cluster centers. The similarity measure is done based on the page visit time using fuzzy intersection and union. Even after preprocessing noise is still present in the web log data.

Olfa Nasraoui, et al. [27] defining the similarity between user session where compute preprocessing and segmentation of web log data is divided into sessions. Preprocessing of web log data and cluster user sessions can achieve using the fuzzy clustering technique. This will affect the clustering result and similarity measures.

Zahid Ansari, et al. [24, 29] explains an existing web usage mining framework. It uses the fuzzy set theoretic approach in preprocessing and in clustering. It improves mining results when compared with crisp approach in preprocessing and clustering. Because the fuzzy approach matches more with real world scenario. It is using the fuzzy c-means algorithm for clustering.

Castellano, et al. [28] aims at clustering website users into different groups and generating session clusters by using a fuzzy C-Means clustering algorithm. The fuzzy set-based approach can solve most of the challenges listed above. FCM needs an initial random selection of clusters. This work focuses on designing "A Spark based Parallel Fuzzy C Median Algorithm for Web Log Big Data". It improves the quality of discovered clusters.

2.3. Review on Apache Spark

• Spark: A variety of tools, technologies, and database systems have been developed as a result of the development of big data platforms to fulfil the growing demands for vital data processing and storage [23]. Apache Spark (Figure 2) is an open-source platform for real-time processing cluster computing that was developed by the Apache Software Foundation. It was built on top of Hadoop MapReduce and extends the MapReduce concept to efficiently use more types of calculations. Unlike other computing clusters like Apache Hadoop, Spark uses a distributed memory model that enables frequently used data and intermediate results to be saved for recurrent processing to achieve high performance [30].

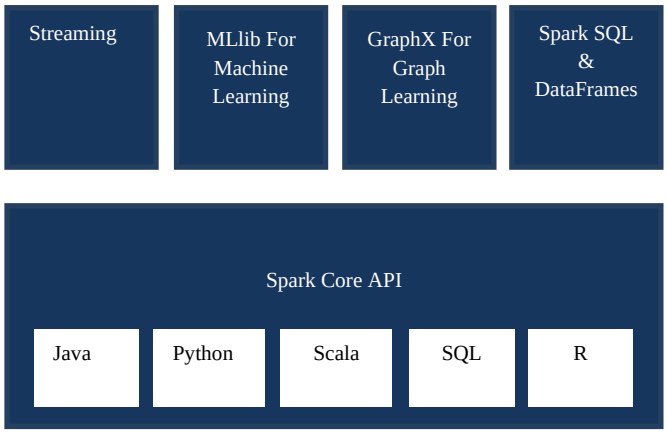

Figure 2. Spark architecture [31]

• RDD: Resilient: Capable of data reconstruction in the case of a failure and fault tolerance (Figure 3).
• Distributed: When data is distributed, it is split up among several nodes in a cluster.
• Dataset: A collection of partitioned data with values is referred to as a "dataset" [32].

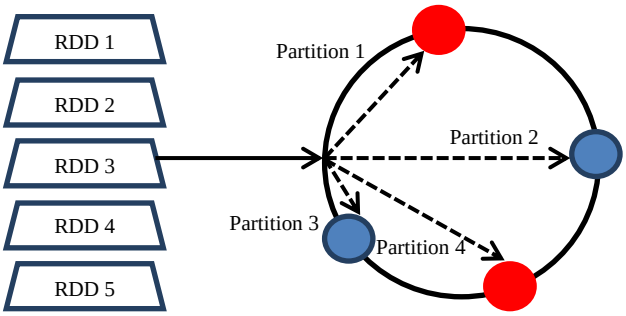

Figure 3. RDD Architecture [32]

2.4. Workflow of RDD

The results of Spark transformations take some time to compute since they are "lazy". Just the operation and the dataset (like a file) it will be applied to are "remembered" by them. When an action is called and the outcome is delivered back to the driver application, the transformations are actually computed. Spark runs more efficiently thanks to its architecture. Each time you

conduct an action on a converted RDD, it is by default recomputed. For instance, Spark would only process and return the result for the first line rather than the complete file if a large file was modified in various ways and presented to the first action. Another choice is to cache or persist an RDD in memory (Figure 4), in which case Spark will maintain the components on the cluster for noticeably faster access the next time you query it.

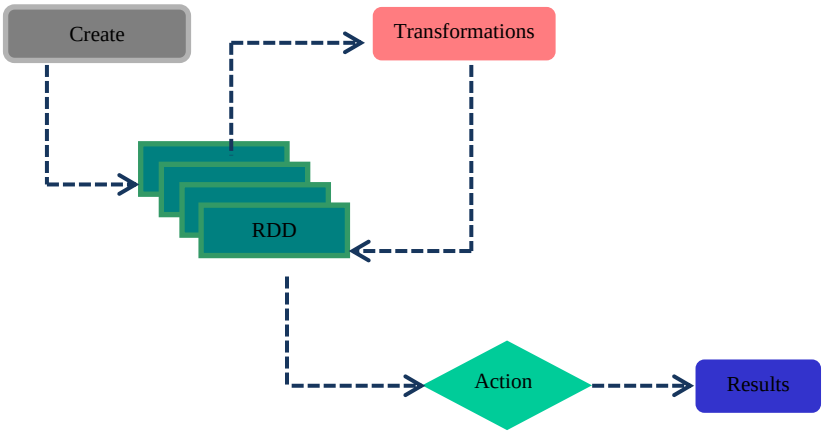

Figure 4. RDD workflow [32]

2.5. Spark Cluster

The Spark Context (SC) object in your main manages the coordination of distinct groups of Spark applications running on a cluster as independent sets of processes.SC may connect to a variety of resource cluster managers that distribute resources among applications, including Mesos/YARN and Spark's standalone cluster (Figure 5) manager. Spark connects to the cluster nodes and obtains executors worker processes that do calculations and store data for your application. The application code is subsequently provided to the executors (JAR or Python files). Finally, SC gives executors duties to finish.

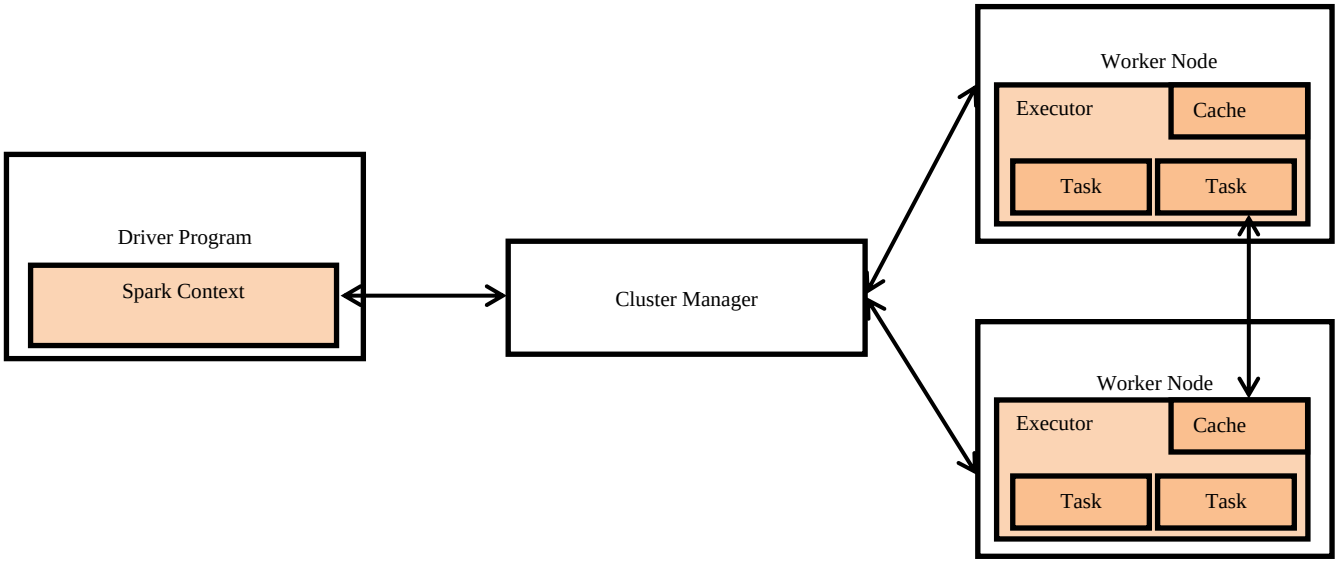

Figure 5. Spark cluster [33]

3. FRAMEWORK AND ALGORITHM

One of the most well-known unsupervised fuzzy clustering techniques is the fuzzy C Means method. A constrained soft clustering algorithm is one of the most well-known unsupervised fuzzy clustering techniques. Identifying separate clusters based on samples' closeness to cluster centers is the main aim of the FCM approach. By minimizing the objective function, the FCM technique determines the appropriate cluster centers that produce good partitioning outcomes.

The criteria for identifying the cluster's center are the total of the distances between locations in different clusters and their centrist address the difficulty of big data processing and analysis for web log massive data, we provide a parallel fuzzy C median technique built on the Spark distributed computing platform. Using cloud data stored in a distributed computing platform, the membership degrees of pixel pointers to various cluster centers and the cluster centers are calculated and updated concurrently for iterative computing (Databricks). The segmented data is then reconstructed using the data from the clustered.

3.1. Framework (Databricks)

Databricks is cloud based big data engineering tool. It is a unified analytics platform developed by the Apache Spark creators. It allows us to quickly launch cloud-optimized Spark clusters. Consider it an all-in-one package for writing code. We can use Spark to generate results without having to worry about the underlying intricacies. It also supports shared Jupyter notebooks, GitHub integration, links to a variety of commonly used tools, and automation monitoring, scheduling, and debugging.

We sign up for the community edition for free. We'll be able to experiment with Spark clusters as a result of this. Depending on the plan, quickly deploy clusters on Central Processing Unit (CPU) and Graphics Processing Unit (GPU) instances on Amazon Web Services (AWS) and Azure for maximum flexibility.

● Step for Overall Process:

Step 1: Create a cluster.

Step 2: Connect a notebook to the cluster and run commands from the notebook.

Step 3: Create a DataFrame from a Databricks dataset using the Python DataFrame API.

Step 4: Create a graph by manipulating the data and show the outcomes

3.2. Create Cluster

The Create button is the simplest approach to start a new cluster: In the sidebar, click Create Icon Create and choose Cluster from the menu. The page Create Cluster displays. The cluster should be given a name and configured.

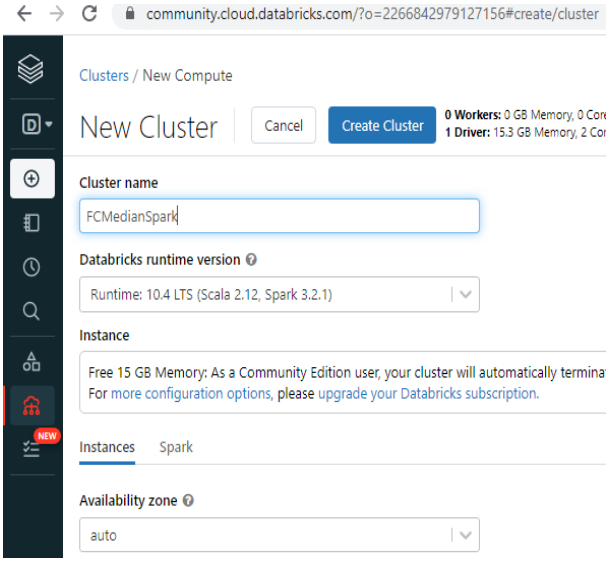

Figure 6. Screenshot of create cluster at data bricks

3.3. Connect a Notebook to the Cluster and Run Commands from the Notebook

A document's web-based user interface is called a notebook that includes executable code, graphics, and narrative prose. This section explains how to utilize and manage notebooks. It also includes about data visualizations, sharing visuals as dashboards, using widgets to parameterize notebooks and dashboards, leveraging notebook workflows to build complicated pipelines and best way for defining classes in notebooks.

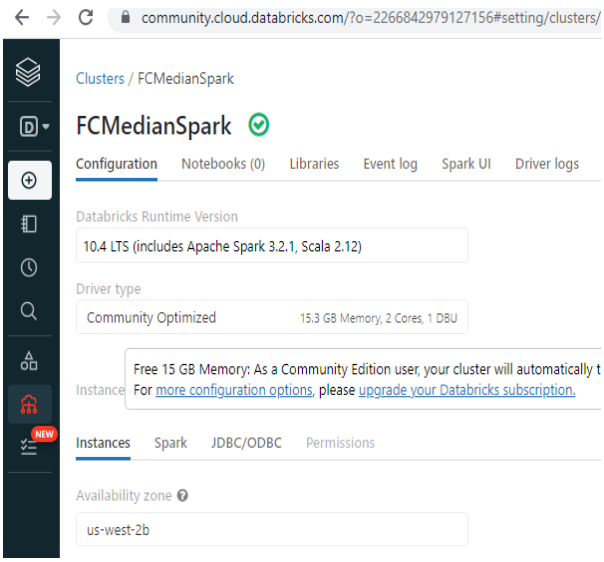

Figure 7. Screenshot of create FCMedianSpark cluster

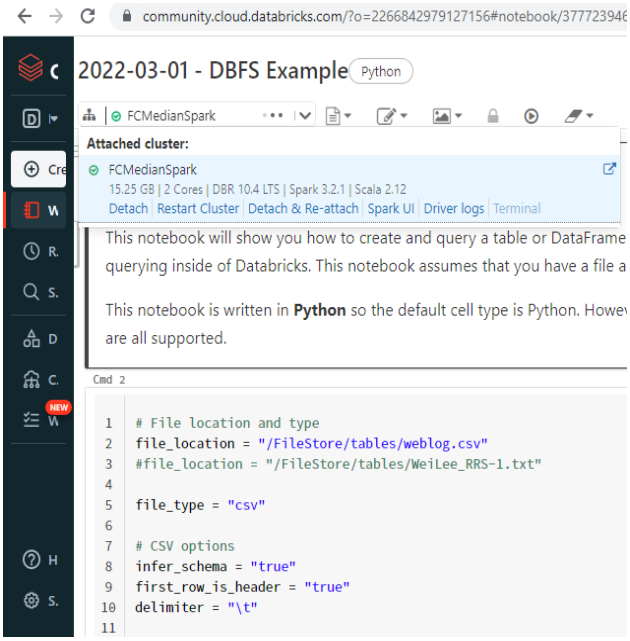

Figure 8. Screenshot of data bricks notebook

3.4. Databricks File System (DBFS)

A cluster-based distributed file system called the Databricks File System (DBFS) can be mounted in a Databricks workspace. Over scalable object storage, DBFS adds an abstraction layer that offers the following benefits:

1. Permit mounting of storage items, which will provide password-free data access.
2. We can link to object storage using directory and file semantics rather than storage URLs.
3. Files are saved to object storage after a cluster is terminated, so we won't lose any data.

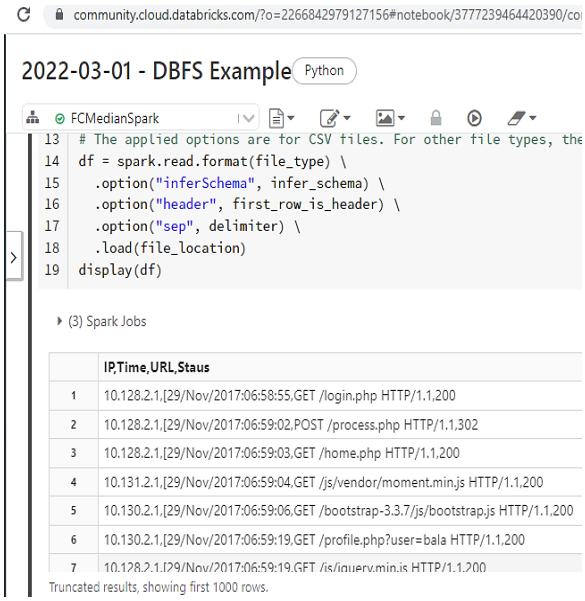

Figure 9. Screenshot of Databricks file system

3.5. Create a DataFrame from a Databricks Dataset Using the Python DataFrame API

A two-dimensional labelled data structure containing several types of columns is known as a DataFrame. A DataFrame can be compared to a spreadsheet, a SQL table, or a dictionary of series objects. To create DataFrame from a Databricks, we used Pyspark. PySpark is the Python API for Apache Spark.

- PySpark: A Python library called PySpark is used to communicate with Apache Spark. It offers both the ability to build Spark applications using Python APIs and the PySpark shell, which enables users to interactively view data in a distributed environment. PySpark supports the vast majority of Spark technologies, including Spark Core, DataFrame, MLlib (Machine Learning), and Spark SQL. The Apache Spark Community developed this Python-Spark integration tool. We may also use PySpark in the Python programming language to communicate with RDDs. They were able to do this because of a library called Py*4j*. The PySpark Shell is a tool that creates the Spark context and links the Python API to the Spark core. Python is now used by the majority of data scientists and analysts due to its large library.

3.6. Proposed Algorithm

- Input: Databrick File System (DBFS)
- Output: Cluster membership

1) Make a cluster in Databrick
2) Connect notebook to the Databrick cluster
3) Read data from the Databrick and produce RDD
4) Update the cluster "*V*" center and distribute "*V*" to a number of nodes
5) Use the formula below to compute the fuzzy membership "U_{ij}" [2].

$$U_{ij} = \frac{\frac{1}{d_{ij}^2}(x_i, v_j)\left(\frac{1}{m-1}\right)}{\sum_{k=1}^{c}\frac{1}{d_{ij}^2}(x_i, v_k)\left(\frac{1}{m-1}\right)} \quad (1)$$

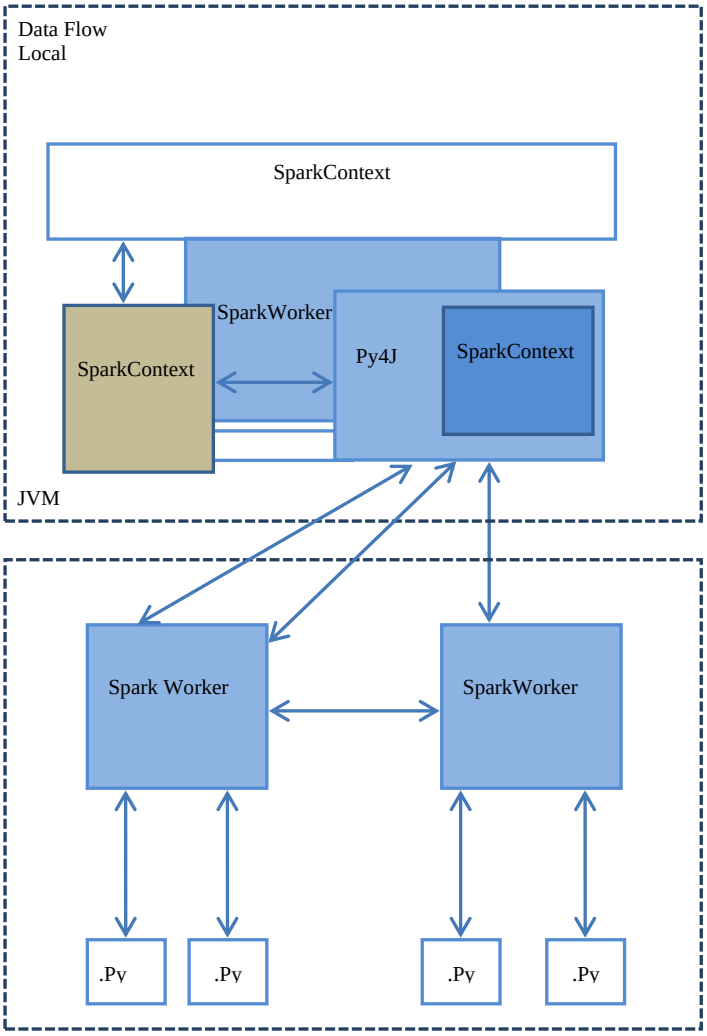

Figure 10. Data flow in PySpark

6) Calculate median through D_{ij} Value for each new cluster centers using below formula [2].

$$D_{ij} = \text{Median}\left\{\left(D_{ij}\left(S_k - S_i\right)^* U_{ij}\right\}; \forall_i \neq k; k = 1...n \quad (2)$$

7) Calculate the fuzzy centers "V_j" using below formula:

$$V_j = \left(\sum_{i=1}^{n}\left(U_{ij}\right)^m x_i\right) / \left(\sum_{i=1}^{n}\left(U_{ij}\right)^m\right); \forall j = 1, 2, 3, ..., c \quad (3)$$

8) The following formula is used to calculate new cluster centers for each iteration:

$$p = \text{Argmin}\left\{\left(D_i{:}n\right); \forall_i{=}1...n\right\} \quad (4)$$

9) Step 5 and 6 will be repeated till the objective function should be minimum "*J*" or

$$\| U_{(k+1)} - U_{(k)} \| < \beta \quad (5)$$

where, the step of iteration is "*k*" and "β" is the criterion for terminating between [0, 1]. $U = \left(U_{ij}\right) n^* V$ is the fuzzy membership matrix.

10) Create a graph showing the cluster centers from $n = 2$ to limit and examine the Validity indices. For each limit, we select cluster centers with less residents.
11) Keep track of the cluster data and output it.

4. RESULTS

The results of different stages are given as follows. Comparing FCM (Fuzzy C Means), FCLM (Fuzzy C Least Median) and the proposed algorithm PFCMS (Parallel Fuzzy C Median using Spark).

4.1. Rand Index

By evaluating all pairs of samples and counting pairs that are assigned in the same or different clusters in the anticipated and true clustering, the Rand Index computes a similarity measure between two clustering. The Rand index ranges from 0 to 1. The Rand index is 1 when the two partitions are exactly aligned. Clustering accuracy is high, and the rand index value is close to one; otherwise, clustering findings are less accurate. Figure 11 depicts the clustering accuracy results of several approaches. It is demonstrated that the suggested PFCMS approach has a higher rand index rate than the FCM and FCLM methods.

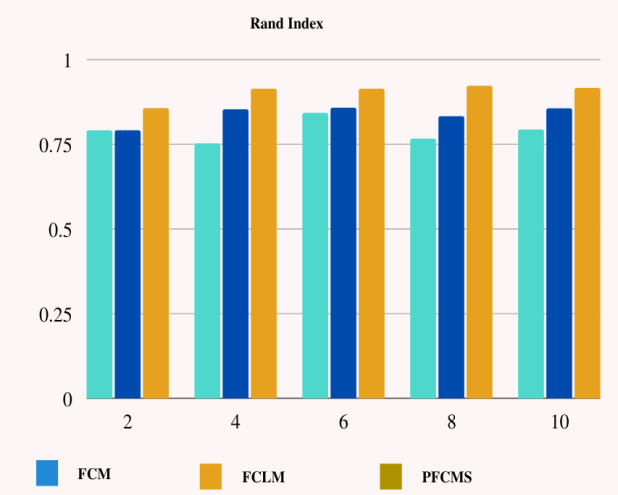

Figure 11. Rand index of FCM, FCLM, PFCMS, respectively

Table 1. Rand Index of algorithms (FCM, FCLM, PFCMS)

Clusters	Rand index (RI) (%)		
	FCM	FCLM	PFCMS
2	0.79	0.79	0.8554
4	0.7512	0.852	0.9123
6	0.8412	0.8562	0.9125
8	0.7652	0.8315	0.921
10	0.7923	0.8545	0.915

4.2. Sum of Squared Error (SSE)

SSE is the most basic and extensively used clustering criterion metric. It assesses the clusters' compactness. Figure 12 shows the results of SSE among clustering algorithms for a user session matrix. The findings reveal that the proposed algorithm has a lower error value than the FCM and FCLM approaches, as shown in Table 2. Cluster analysis is a statistical technique for determining which consumers (or respondents) are the "best fit" for a certain market segment (cluster). The lower the SSE, the more similar the market segment's consumers are.

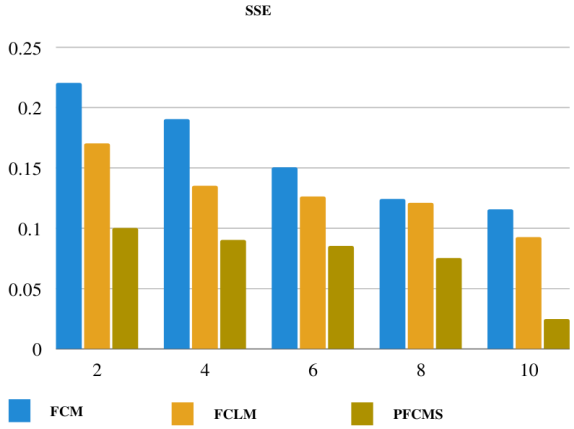

Figure 12. SSE of PFCMS, FCLM, FCM, respectively

Table 2. SSE of clustering algorithms (FCM, FCLM, PFCMS)

Clusters	(SSE) (%)		
	FCM	FCLM	PFCMS
2	0.22	0.17	0.1
4	0.19	0.1348	0.09
6	0.15	0.1259	0.085
8	0.1238	0.1205	0.075
10	0.111	0.0922	0.0248

5. CONCLUSION

This research proposes a new approach for clustering web user transactions based on fuzzy C Means in fuzzy environments. Similar user navigation patterns are grouped together in this method. Using Apache Spark to discover the member with the greatest membership within a cluster improves the FCM and offers a concurrent Fuzzy C Median technique. The membership degrees of pixel points to various cluster centers and the cluster centers are calculated and updated in parallel for iterative computing once the cloud data is created and saved in distributed computing platforms. The results reveal that Spark-based parallel fuzzy C-median algorithm achieves a good performance on distributed computing node. The algorithm was tested and analyzed, and it was discovered to be a better way for clustering than existing algorithms.

REFERENCES

[1] T.H. Sardar, Z. Ansari, "MapReduce-Based Fuzzy C-Means Algorithm for Distributed Document Clustering", The Institution of Engineers, Vol. 103, Issue 3, pp. 131-142, Kolkata, India, 2022.

[2] H.B. Guliyev, "Fuzzy Probabilistic Model for Managing the Modes of Networks with Renewable Energy Sources", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 46, Vol. 13, No. 1, pp. 46-50, March 2021.

[3] M.A. Mallik, et al., "An Efficient Fuzzy C-Least Median Clustering Algorithm", IOP Conf. Ser., Mater. Sci. Eng. Vol. 1070, Issue 1, p. 012050, Tamil Nadu, India, 2021.

[4] M. Bendechachea, A.K. Tarib, M.T. Kechadiaa, "Insight Centre for Data Analytics", University College Dublin, Ireland University A-Mira of Bejaia, Algeria "Parallel and Distributed Clustering Framework for Big Spatial Data Mining", Article in International Journal of

Parallel Emergent and Distributed Systems, Vol. 34, Issue 6, pp. 671-689, March 2018.
[5] R. Cooley, B. Mobasher, J. Srivastava, "Web Mining: Information Andpattern Discovery on the World Wide Web", The Ninth IEEE International Conference, pp. 558-567, London, UK, November 1997.
[6] M.A. Mallik, et al., "A Survey on Parallel Clustering Techniques for Big Data Framework", The 2nd Global Conference on Artificial Intelligence and Applications (GCAIA 2021), CRC Press, Taylor and Francis, pp. 49-56, Jaipur, India, 2022.
[7] A. Sinha, P.K. Jana, "A Novel K-Means Based Clustering Algorithm for Big Data", IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1875-1879, 2016.
[8] C. Sreedhar, N. Kasiviswanath, P.C. Reddy, "Clustering Large Datasets using K-Means Modified Inter and Intra Clustering (KM-I2C) in Hadoop", Journal of Big Data, Vol. 4, No. 1, p. 27, 2017.
[9] N. Akthar, M.V. Ahamad, S. Khan, "Clustering on Big Data Using Hadoop MapReduce", The 2015 IEEE International Conference on Computational Intelligence and Communication Networks (CICN), pp. 789-795, 2015.
[10] A. Sinha, P.K. Jana, "A Novel K-Means Based Clustering Algorithm for Big Data", The 2016 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1875-1879, 2016.
[11] M.A. Ben Haj Kacem, C.E. Ben N'cir, N. Essoussi, "One-Pass MapReduce-Based Clustering Method for Mixed Large-Scale Data", Journal of Intelligent Information Systems, pp. 1-18, 2017.
[12] M.O. Shafiq, E. Torunski, "A Parallel K-Medoids Algorithm for Clustering Based on MapReduce", The 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 502-507, 2016.
[13] M.A. Ben Haj Kacem, C.E. Ben N'cir, N. Essoussi, "MapReduce-Based K-Prototypes Clustering Method for Big Data", The 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1-7, 2015.
[14] S.A. Ludwig, "MapReduce-Based Fuzzy C-Means Clustering Algorithm: Implementation and Scalability", International Journal of Machine Learning and Cybernetics, Vol. 6, No. 6, pp. 923-934, 2015.
[15] M.S. Hidri, M.A. Zoghlami, R. Ben Ayed, "Speeding up the Large-Scale Consensus Fuzzy Clustering for Handling Big Data", Fuzzy Sets and Systems, 2017.
[16] Q. Zhang, Z. Chen, "A Weighted Kernel Possibilistic C-Means Algorithm Based on Cloud Computing for Clustering Big Data", International Journal of Communication Systems, Vol. 27, No. 9, pp. 1378-1391, 2014.
[17] Q. Zhang, L.T. Yang, Z. Chen, P. Li, "PPHOPCM: Privacy-Preserving High-Order Possibilistic C-Means Algorithm for Big Data Clustering with Cloud Computing", IEEE Transactions on Big Data, No. 99, pp. 1-11, May 2017.
[18] R. Hu, W. Dou, J. Liu, "ClubCF: A Clustering-Based Collaborative Filtering Approach for Big Data Application", IEEE Transactions on Emerging Topics in Computing, Vol. 2, No. 3, pp. 302-313, 2014.
[19] V. Subramaniyaswamy, V. Vijayakumar, R. Logesh, V. Indragandhi, "Unstructured Data Analysis on Big Data Using MapReduce", Procedia Computer Science, Vol. 50, pp. 456-465, 2015.
[20] P. Sachar, V. Khullar, "Social Media Generated Big Data Clustering Using Genetic Algorithm", The 2017 IEEE International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, pp. 1-6, 2017.
[21] J. Karimov, M. Ozbayoglu, "High Quality Clustering of Big Data and Solving Empty-Clustering Problem with an Evolutionary Hybrid Algorithm", The 2015 IEEE International Conference on Big Data (Big Data), pp. 1473-1478, 2015.
[22] Y. Yang, F. Teng, T. Li, H. Wang, H. Wang, Q. Zhang, "Parallel Semi-Supervised Multi-Ant Colonies Clustering Ensemble Based on MapReduce Methodology", IEEE Transactions on Cloud Computing, Vol. 6, No.1, pp. 1-12, 2015.
[23] M. Sais N. Rafalia J. Abouchabaka, "Enhancements and Intelligent Approach to Optimize Big data Storage and Management: Random Enhanced HDFS (REHDFS) and DNA Storage", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 50, Vol. 14, No. 1, pp. 196-203, March 2022.
[24] Z. Ansari, M.F Azeem, A.V. Babu, W. Ahmed, "A Fuzzy Clustering based Approach for Mining Usage Profiles from Web Log Data", International Journal of Computer Science and Information Security (IJC-SIS), Vol. 9, No. 6, Vol. 9, pp. 70-79, June 2011.
[25] G. Castellano, F. Mesto, M. Minunno, M.A. Torsello, "Web User Profiling Using Fuzzy Clustering", WILF (F. Masulli, S. Mitra, G. Pasi, eds.), Lecture Notes in Computer Science, Vol. 4578, pp. 94-101, Springer, 2007.
[26] O. Nasraoui, H. Frigui, R. Krishnapuram, A. Joshi, "Extracting Web User Profiles Using Relational Competitive Fuzzy Clustering", International Journal on Artificial Intelligence Tools, Vol. 9, No. 4, pp. 509-526, 2000.
[27] G. Castellano, A.M. Fanelli, M.A. Torsello, "Mining Usage Profiles Fromaccess Data Using Fuzzy Clustering", The 6th WSEAS International Conference on Simulation, Modelling and Optimization (SMO), pp. 157-160, Stevens Point, Wisconsin, USA, 2006.
[28] Z. Ansari, M.F. Azeem, A.V. Babu, A. Waseem, "A Fuzzy Approach for Feature Evaluation and Dimensionality Reduction to Improve the Quality of Web Usage Mining Results", International Journal on Advanced Science, Engineering and Information Technology, Vol. 2, No. 6, pp. 67-73, 2012.
[29] Z. Ansari, A. Babuy, W. Ahmed, M. Azeemz, "A Fuzzy Set Theoretic Approach to Discover User Sessions from Web Navigational Data", Recent Advances in Intelligent Computational Systems (RAICS), pp. 879-884, September 2011.

[30] S.R. Gallego, S. Garcia, J.M. Benitez, F. Herrera, "A Distributed Evolutionary Multivariate Discretizer for Big Data Processing on Apache Spark", Swarm Evol. Comput., Vol. 38, pp. 240-250, February 2018.
[31] https://data-flair.training/blogs/apache-sparkeco system-components/
[32] www.edureka.co/blog/spark-architecture/
[33] https://intellipaat.com/blog/tutorial/sparktutorial/ spark-architecture/

BIOGRAPHIES

Moksud Alam Mallik was born in Howrah, Kolkata, India on June 20, 1985. He completed his B.Tech. (CSE) in 2005 from Maulana Abul Kalam Azad University of Technology, West Bengal, India, MTech. (CSE) in 2014 from Visvesvaraya Technological University (VTU), Karnataka, India, both with distinction and awarded Ph.D. (CSE) in 2022 from International Islamic University Malaysia (IIUM), Kualalampur, Malaysia. He has more than 15 years of teaching and industrial experience in reputed engineering colleges and renowned MNCs. He has 4 books chapter and 11 research publications in reputed Journals (Scopus, WoS), conferences, proceedings of which is published by Springer and IEEE. His research interests broadly lie in parallel clustering algorithm, big data analytics, data mining, machine learning and data science.

Nurul Fariza Zulkurnain was born in Malaysia. She obtained her Ph.D. in Computer Science from University of Manchester, UK in 2012. She is currently an Associate Professor at Electrical and Computer Engineering Department, Kulliyyah of Engineering, International Islamic University Malaysia. She has 19 years of teaching and research experience. As a part of the Software Engineering Research Group, her research interests are in the area of big data, data mining, machine learning and artificial intelligence.

Mohammed Khaja Nizamuddin was born in Telangana, India on March 17, 1979. He completed his B.E. (CSE) from Osmania University, Hyderabad, India in 2002, MTech. (CS) from Jawaharlal Nehru Technological University Hyderabad, Hyderabad, India in 2008, both with distinction and awarded Ph.D. (CSE) from Rayalaseema University, Pasupula, India in 2017. He has more than 19 years of teaching experience in reputed engineering colleges. He has 6 National Patents and 13 research publications in reputed Journals (Scopus, WoS), conferences, proceedings of which is published by Springer and IEEE. His area of research is database transaction management, cyber security and machine learning.

Rashal Sarkar was born in Kakripara, Assam, India on June 18, 1983. He received the engineer degree from Biju Patnaik University of Technology, Orissa, India, M.Tech. (CSE) degree from R.V College of Engineering, Bangalore, India and Ph.D. from Himalayan Garhwal University, Uttarakhand, India. He has teaching and administrative experience of more than 16 years at college and university level. Currently, he is an Associate Professor in Department of Computer Science, University of Science and Technology, Meghalaya, India. He has more than 10 publications in reputed international journals and conferences. His expertise is in data mining, data science, machine learning, and artificial intelligent.

Aboosalih Kakkat Chalil was born in Malappuram, Kerala, India in Nov. 22, 1980. He obtained his B.E. (CSE) from Anna University, Chennai, Tamilnadu, India in 2006, M.Techz. (CSE) from National Institute of Technology (NIT) Calicut, Kerala, India in 2015 both with first class. He has more than 8 years of teaching and industrial experience in software firms and engineering colleges. He has 3 research publications in reputed journals (Scopus, Springer). His research lies in clustering, fuzzy clustering, data mining, machine learning and data science.