ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2018, Vol. 07, No. 2, 15-22

Use a cluster approach to organize and analyze data inside the cloud

N. Boyko, P. Mykhailyshyn, Y. Kryvenchuk

Lviv Polytechnic National University, Lviv, Uktaine; e-mail: nataliya.i.boyko@lpnu.ua

Received February 19.2018: accepted June 02.2018

Abstract. In this paper, description of a concept of cloud storage is offered. Cloud data storage is a model of an online storage where the data is being stored in multiple, divided between network servers that are provided for clients' usage, mostly by a third-party company. The majority of the cloud storages (as opposed to file-exchangers) are offering almost boundless set of functions for free, by only limiting the size of the available storage (mostly a couple of gigabytes). Integrated data mining is being used for extracting potentially useful information from unprocessed data. The methods of data analysis are quite important with cloud computing. The implementation of the methods of integrated data mining inside the cloud will let the users receive the helpful information from non-structured or half-constructed web data sources. The main purpose of this work is to organize huge diverse data coming from different sources into clusters, depending on the type of data.

Keywords: information process, cloud computing, information system, cloud storage, data cloud, cloud service, cluster approach, clustering.

INTRODUCTION

Cloud computing is a fairly new technology that is ensuring computing resources to be presented as servers, such as infrastructure, storage for keeping data, the platforms for developing the applications, the software, etc. Cloud computing becoming more and more popular, and as of today are being widely used in different areas of IT, and even more. Large amount of data is being stored on cloud, and that data needs to be received by the user in an efficient way. Uploading information from the cloud takes a lot of time since the data is not being stored in organized sequence. Therefore, the intellectual data mining plays an important part in cloud computing. We are now able to integrate intellectual data mining and cloud computing (IDMCC) that are ensuring the maneuvering and quick access to this technology. An integration is supposed to be strong to the point of being capable to comprehend the growing amount of data, and will also help to effectively analyze huge amount of it. In this article we will provide you with the short description of cloud computing and methods of cloistering, and later on about intellectual mining of that data inside the cloud. In the current work there is a proposal of a model that engages hierarchical algorithm of data cloistering inside the cloud storages for differentiating the data that are based on the information downloaded by the different end users. [4, 7].

ORGANIZATION AND ANALYZING THE DATA INSIDE THE CLOUD

Cloud computing are becoming very popular, and many IT-giants such as Google, Microsoft, IBM have been using them for awhile, developing new features in this area and establishing their own infrastructure using this very cloud computing. A cloud can be seen as an infrastructure that provides with the resources/services through the internet. The benefits of cloud computing in comparison with the traditional computing hardware include: maneuvering, lower cost of entering, scaling and independence regarding the location. The main features of cloud computing are: self-maintenance (when needed), access to network, collaboration of the resources, fast versatility, measured service. Cloud computing is the technology of the new generation that can substitute the other already existing technology, since it allows it's clients to use the services without being worried about the infrastructure, installing, configuring, etc., and offers them to pay only for the services that are being used. Cloud can be a cloud storage, cloud of computing of data cloud. Cloud storage provides with the services of storing (service based on block-file) that support, manage and execute reserve copying of data with the huge distance in between that gives the users an ability to access that data. The main benefit of cloud storage is the ability of data being preserved virtually. Data cloud provides with the services of managing the data, therefore computing cloud provides with the calculating services. Nowadays the architecture of the cloud is structured in top and modern data centers. It includes IAAS, PAAS, and SAAS. The following scheme is illustrating hierarchical representation of the cloud calculating.

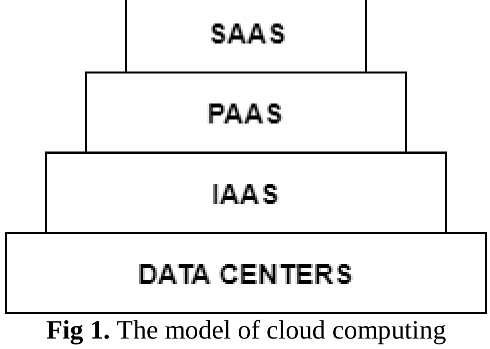

Fig 1. The model of cloud computing

DATA-centers: these are the base of cloud computing, that provide with the apparatus insurance for cloud to work. This is the centralized storage for saving and managing the data.

IAAS: built above the data-centers, **IAAS** layer provides with the calculating infrastructure, such as storing, network connection of the servers from the centers of data processing, firewall of the IT address, etc. Examples: Amazon EC2, Windows Azure, Google Compute Engine.

PAAS: provides with the software for the end users as the service on demand. SAAS getting rid of the necessity of installing the setup and starting the applications on separate computes. Examples: Google Map, Google applications, Microsoft 365 [2, 5].

Integrated data mining is being used for extracting potentially useful information from unprocessed data. The methods of data analysis are quite important with cloud computing. The implementation of the methods of integrated data mining inside the cloud will let the users receive the helpful information from non-structured or half-constructed web data sources. The users need to analyze and highlight important data patterns that are being stored inside the large data centers. In this way the instruments of integrated data mining play an important role when it comes to cloud computing. The application of contemporary algorithms appeared to be ineffective in the cloud environment. They are not suitable for large distributed data bases, as it takes a lot of time to execute. There are many algorithms that can handle large databases but using a large amount if internal memory is the biggest concern. Using a cloud to process and store a database can solve this problem since it can handle additional memory needs very easily. The main purpose of this work is to organize huge diverse data coming from different sources into clusters, depending on the type of data. This will ensure fast data extraction from huge cloud data centers [10].

THE USAGE OF CLUSTER APPROACH

1. Clustering

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to each other within a single cluster and are not homogeneous to objects inside the other clusters. This is quite a useful method for detecting the distribution of data and the structure of the outcoming data.

2. Clustering Types

The methods of clustering can be classified in the following way:

– **Partitioning Method:** This is the simplest and most fundamental version of cluster analysis that distributes set objects to multiple exclusive groups or clusters. They come in two types: k-means (k-value) and k-medoids.

– **Hierarchical Method:** It works by grouping data objects into a cluster tree. The algorithm iteratively divides the database into smaller subsets, until some completion condition is satisfied. Hierarchical clustering methods can be classified as sintering and separating. **Agglomeration** hierarchical clustering is a bottom-up strategy, it begins by placing each object in its own cluster, and then combining these atomic clusters into larger and larger clusters until all the objects are in the same cluster or until meeting a certain condition. **Split** hierarchical clustering is a top-down strategy and does the opposite of agglomeration hierarchical clustering, starting with all objects in one cluster.

– **Density Based Method Clustering**: The method is based on density, and its general idea is to continue to build the base cluster until the data density exceeds a certain threshold. It consists of basic functions such as: detecting clusters of arbitrary shape, noise handles and one scan. There are basically three types: DBSCAN, OPTICS, and DENCLUE.

– **Grid Based Clustering**: This method uses a multi-level grid data structure. This is the quantization of space objects on the finite number of cells that form the mesh structure in which all clustering operations are executed. STING is a multi-stage mesh technique in which the spatial area is divided into rectangular cells. Mostly there are several levels of such rectangular cells corresponding to different levels of permission, and these cells form a hierarchical structure. CLIQUE (clustering in QUEST) was the first algorithm proposed for increasing the number of clustering subspace measurements in a multidimensional space. In the growth space of measurements, the process of clustering begins in a one-dimensional space and grows up to multidimensional spaces.

– **Model Based Clustering** hypothesizes the model for each of the clusters and finds the best data match for the given model. Such methods are often based on the assumption that data is generated using a mixture of basic probabilistic distributions. There are 3 types of those: Expectation Maximization, Conceptual Clustering, and Neural Networks Approach. The Expectation Maximization (EM) algorithm is a popular iterative algorithm that can be used to find parameter estimates. It can be considered as a continuation of the paradigm k-means (k-value), which assigns an object in a cluster with which it resembles the most.

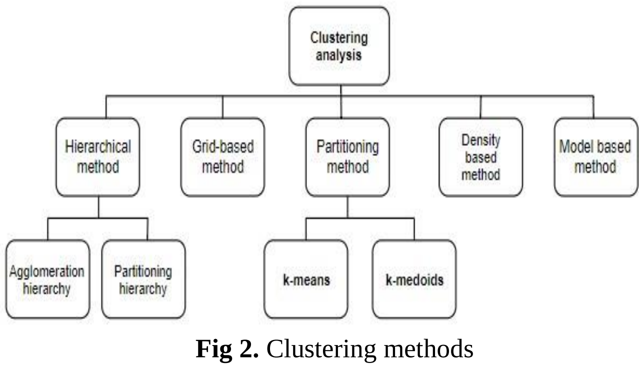

Fig 2. Clustering methods

A COMPARISON OF THE MOST COMMON CLUSTERING ALGORITHMS

Hierarchical Clustering Algorithms

The operation of a hierarchical clustering algorithm is illustrated using the two-dimensional data set in Figure 3.

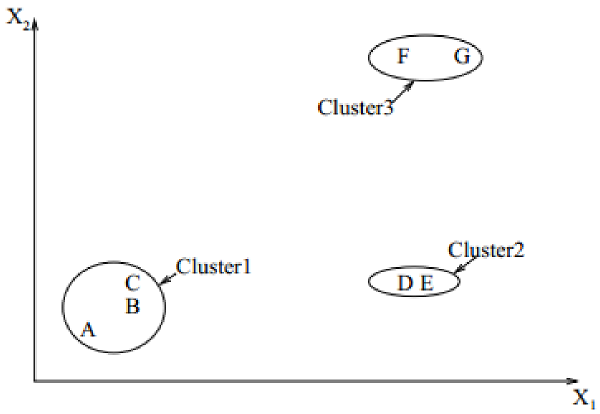

Fig 3. Points falling in three clusters

This figure depicts seven patterns labeled A, B, C, D, E, F, and G in three clusters. A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. A dendrogram corresponding to the seven points in Figure 3 (obtained from the single-link algorithm [7]) is shown in Figure 4.

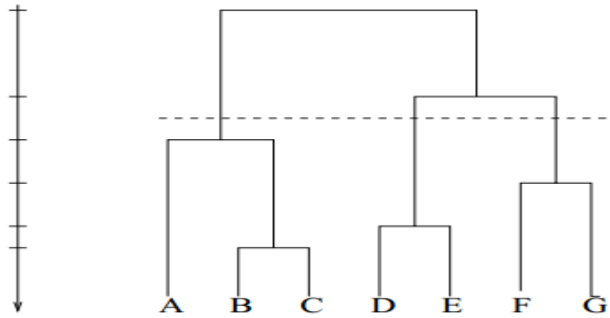

Fig 4. The dendrogram obtained using the single-link algorithm

The dendrogram can be broken at different levels to yield different clusterings of the data. Most hierarchical clustering algorithms are variants of the single-link [9], complete-link [3], and minimum-variance [8] algorithms.

Of these, the single-link and complete-link algorithms are most popular. These two algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second). In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-link algorithm produces tightly bound or compact clusters [9]. The single-link algorithm, by contrast, suffers from a chaining effect [4]. It has a tendency to produce clusters that are straggly or elongated. There are two clusters in Figures 12 and 13 separated by a “bridge” of noisy patterns. The single-link algorithm produces the clusters shown in Figure 5, whereas the complete-link algorithm obtains the clustering shown in Figure 6.

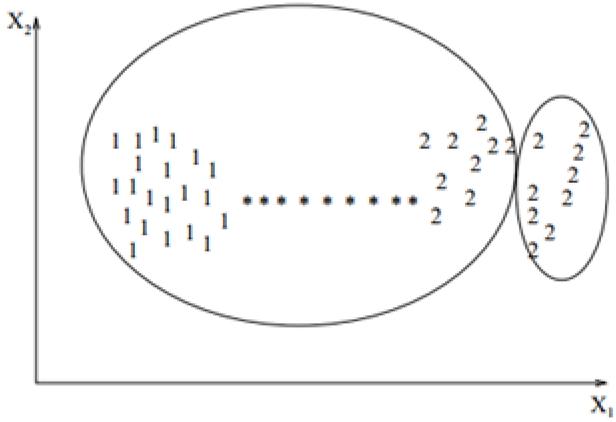

Fig 5. A single-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

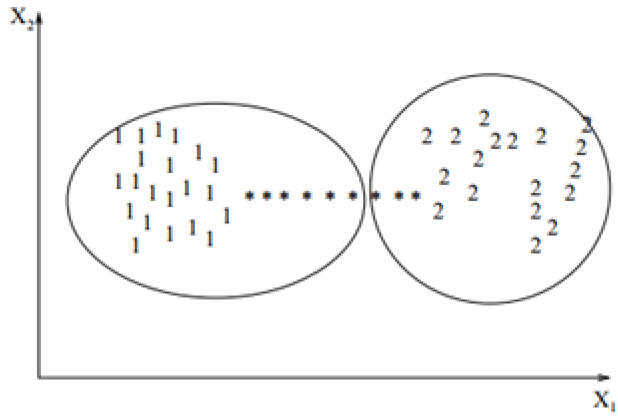

Fig 6. A complete-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

The clusters obtained by the complete-link algorithm are more compact than those obtained by the single-link algorithm; the cluster labeled 1 obtained using the single-link algorithm is elongated because of the noisy patterns labeled “*”. The single-link algorithm is more versatile than the complete-link algorithm, otherwise. For example, the single-link algorithm can extract the concentric clusters shown in Figure 7, but the complete-link algorithm cannot.

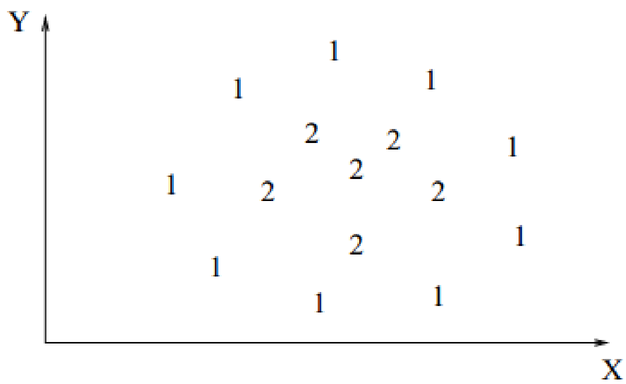

Fig 7. Two concentric clusters

However, from a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications than the single-link algorithm [1].

- Agglomerative Single-Link Clustering Algorithm:
  (1) Place each pattern in its own cluster. Construct a list of interpattern distances for all distinct unordered pairs of patterns, and sort this list in ascending order.
  (2) Step through the sorted list of distances, forming for each distinct dissimilarity value $\mathbf{d_k}$ a graph on the patterns where pairs of patterns closer than $\mathbf{d_k}$ are connected by a graph edge. If all the patterns are members of a connected graph, stop. Otherwise, repeat this step.
  (3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by completely connected components in the corresponding graph.
- Agglomerative Complete-Link Clustering Algorithm:
  (1) Place each pattern in its own cluster. Construct a list of interpattern distances for all distinct unordered pairs of patterns, and sort this list in ascending order.
  (2) Step through the sorted list of distances, forming for each distinct dissimilarity value $\mathbf{d_k}$ a graph on the patterns where pairs of patterns closer than $\mathbf{d_k}$ are connected by a graph edge. If all the patterns are members of a connected graph, stop.
  (3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by completely connected components in the corresponding graph.

Hierarchical algorithms are more versatile than partitional algorithms. For example, the single-link clustering algorithm works well on data sets containing non-isotropic clusters including well- well-separated, chain-like, and concentric clusters, whereas a typical partitional algorithm such as the k-means algorithm works well only on data sets having isotropic clusters [8]. On the other hand, the time and space complexities [9] of the partitional algorithms are typically lower than those of the hierarchical algorithms. It is possible to develop hybrid algorithms [10] that exploit the good features of both categories.

- Agglomerative hierarchical clustering algorithm:
  (1) Compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.
  (2) Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.
  (3) If all patterns are in one cluster, stop. Otherwise, go to step 2.

Based on the way the proximity matrix is updated in step 2, a variety of agglomerative algorithms can be designed. Hierarchical divisive algorithms start with a single cluster of all the given objects and keep splitting the clusters based on some criterion to obtain a partition of singleton clusters.

PARTITIONAL ALGORITHMS

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure, such as the dendrogram produced by a hierarchical technique. Partitional methods have advantages in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive. A problem accompanying the use of a partitional algorithm is the choice of the number of desired output clusters. A seminal paper [2] provides guidance on this key design decision. The partitional techniques usually produce clusters by optimizing a criterion function defined either locally (on a subset of the patterns) or globally (defined over all of the patterns). Combinatorial search of the set of possible labelings for an optimum value of a criterion is clearly computationally prohibitive. In practice, therefore, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering.

The most intuitive and frequently used criterion function in partitional clustering techniques is the squared error criterion, which tends to work well with isolated and compact clusters. The squared error for a clustering C of a pattern set S (containing K clusters) is:

$$e^2(C,S) = \sum_{j=1}^{k}\sum_{i=1}^{n_j}\left|x_i^{(j)} - c_j\right|^2, \qquad (1)$$

where $X^{(j)}_i$ is the *i*-th pattern belonging to the *j*-th cluster and c_j is centroid *j*-th cluster.

The **k-means** is the simplest and most commonly used algorithm employing a squared error criterion [7]. It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centers until a convergence criterion is met (e.g., there is no reassignment of any pattern from one cluster to another, or the squared error ceases to decrease significantly after some number of iterations). The **k-means** algorithm is popular because it is easy to implement, and its time complexity is $\boldsymbol{O(n)}$, where $\boldsymbol{n}$ is the number of patterns. A major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. Figure 8 shows seven two-dimensional patterns. If we start with patterns A, B, and C as the initial means around which the three clusters are built, then we end up with the partition {{A}, {B, C}, {D, E, F, G}} shown by ellipses.

Several variants [3] of the k-means algorithm have been reported in the literature. Some of them attempt to select a good initial partition so that the algorithm is more likely to find the global minimum value. Another variation is to permit splitting and merging of the resulting clusters. Typically, a cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when the distance between their centroids is below another pre-specified threshold. Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified. The well-known ISODATA [5] algorithm employs this technique of merging and splitting clusters. If ISODATA is given the "ellipse" partitioning shown in Figure 8 as an initial partitioning, it will produce the optimal three-cluster partitioning.

ISODATA will first merge the clusters {A} and {B,C} into one cluster because the distance between their centroids is small and then split the cluster {D,E,F,G}, which has a large variance, into two clusters {D,E} and {F,G}.

Another variation of the k-means algorithm involves selecting a different criterion function altogether. The dynamic clustering algorithm (which permits representations other than the centroid for each cluster) was proposed in Diday [1, 3], and Symon [7] and describes a dynamic clustering approach obtained by formulating the clustering problem in the framework of maximum-likelihood estimation. The regularized Mahalanobis distance was used in Mao and Jain [6] to obtain hyperellipsoidal clusters.

The best-known graph-theoretic divisive clustering algorithm is based on construction of the minimal spanning tree (MST) of the data [1], and then deleting the MST edges with the largest lengths to generate clusters. Figure 9 depicts the MST obtained from nine two-dimensional points.

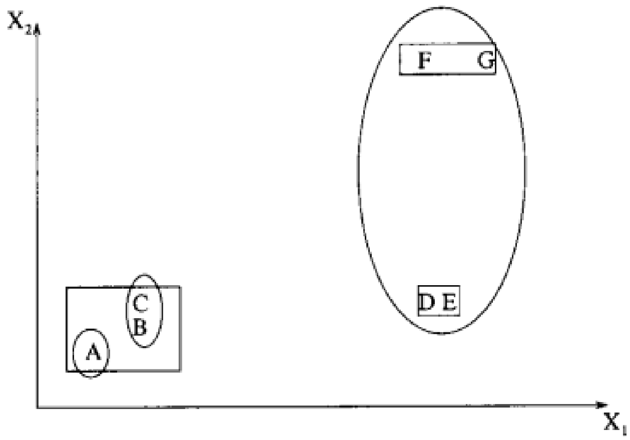

Fig 8.The **K-means** algorithm is sensitive to the initial partition.

The squared error criterion value is much larger for this partition than for the best partition {{A, B, C}, {D, E}, {F, G}} shown by rectangles, which yields the global minimum value of the squared error criterion function for a clustering containing three clusters. The correct three-cluster solution is obtained by choosing, for example, A, D, and F as the initial cluster means.

- <u>Squared error clustering method:</u>
  (1) Select an initial partition of the patterns with a fixed number of clusters and cluster centers.
  (2) Assign each pattern to its closest cluster center and compute the new cluster centers as the centroids of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.
  (3) Merge and split clusters based on some heuristic information, optionally repeating step 2.

- <u>The k-Means Clustering Algorithm:</u>
  (1) Choose k cluster centers to coincide with k randomly-chosen patterns or k randomly defined points inside the hypervolume containing the pattern set.
  (2) Assign each pattern to its nearest cluster center.
  (3) Recompute the cluster centers, using current cluster membership.
  (4) If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centers, or minimal decrease in squared error.

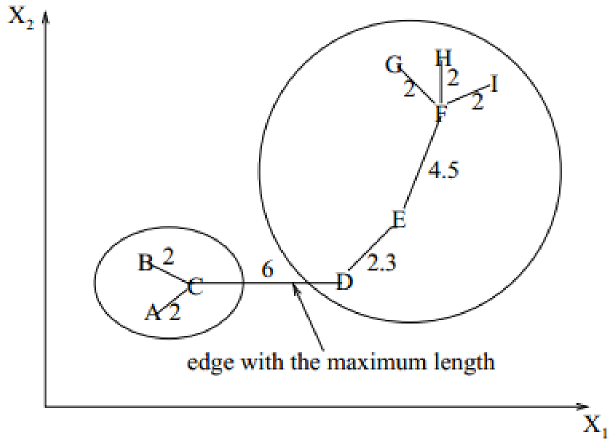

Fig 9. Using MST (minimal spanning tree) to form clusters

By breaking the link labeled CD with a length of 6 units (the edge with the maximum Euclidean length), two clusters ({A, B, C} and {D, E, F, G, H, I}) are obtained. The second cluster can be further divided into two clusters by breaking the edge EF, which has a length of 4.5 units. The hierarchical approaches are also related to graph-theoretic clustering. Single-link clusters are subgraphs of the minimum spanning tree of the data [5] which are also the connected components [2].

FEATURES OF CLOUD COMPUTING

One of the services offered to us by cloud is the storage of data. Previously, prior to the concept of cloud computing, important industrial data was usually stored on different media. Starting from music files to images and secret documents, the cloud invisibly backs up all files and folders and eliminates the necessity for endless and resource-intensive searches for additional storage space. When a user has a large amount of data that needs to be placed somewhere, the cloud of accumulation (storage) greatly helps him, saving money on new accumulating devices and preventing the removal of old data. Therefore, many companies use cloud environments and their storage service. These organizations pay for the amount of space they use in the cloud. Cloud storage is convenient and cost effective. It works by storing files on the server somewhere on the Internet, not on a local hard drive. This allows you to back up, synchronize, and access data on multiple devices as long as the users have an access to the Internet.

Many different studies have been conducted to improve cloud computing. Different algorithms of data mining have been applied in different ways to effectively manage a huge amount of data in the cloud. Current work in this area is Bhupendra Panchal and R.K Kapoor, who proposed clustering and caching as a methodology that can improve the performance of cloud computing. The basic idea is to make exact copies of the data in each data center, so that even if one processing center goes down, everything in the second data center will cluster from the first one and the data becomes available. Kashish Ara Shaki and Mansaf Alam offered an approach that provides cloud data management through clustering and uses K-medians as a clustering method. A.Mahendiran proposed the use of clustering algorithm for k-values in cloud computing for large databases. Kriti Srivastava proposed the implementation of an agglomeration hierarchical clustering algorithm in order to take advantage of such things as scalability, elasticity and the ability to process large amounts of data.

PROPOSED MODE

Cloud storage consists of data incoming from different sources of different types. Traditional data management methods are intended to handle traditional data that were of a certain type and limited in scope, but the available data is currently enormous and heterogeneous (of a different "nature"), which can be both structured, and unstructured. Thus, traditional methods are not able to cope with the data management requirements in the cloud. Data analysis methods and cloud computing help business organizations maximize profits and reduce costs in a variety of ways. One of the most important cloud storage tasks is quick access to data. Any organization that uses cloud storage expects it to be able to efficiently retrieve information in the shortest time possible. But this expectation is not being satisfied because the data in the cloud is unorganized and, thus, it takes some time to access it. There are various end user / client cloud environments. Each client uses a certain amount of paid space in the cloud storage to store their large amount of data.

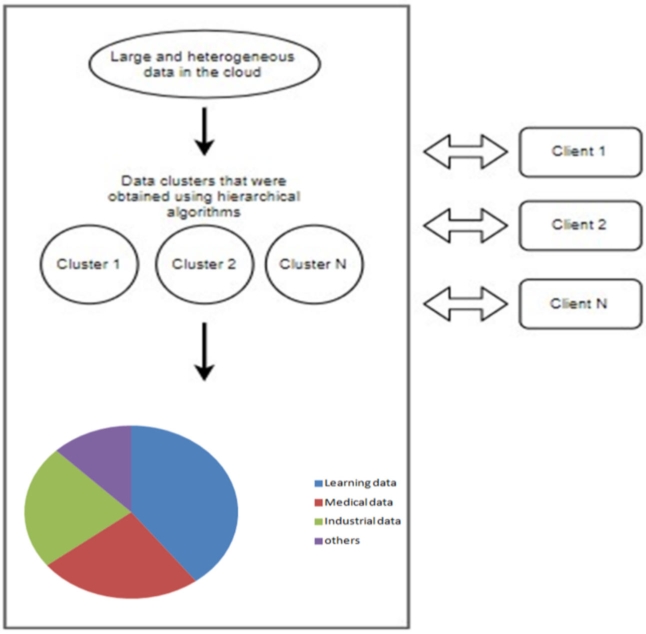

Fig 10. Proposed model

The purpose of our proposed model is to apply a hierarchical clustering algorithm in the cloud storage data centers to form clusters of data based on data types that have been downloaded to the cloud by different organizations. For instance: data from various organizations such as the colleges, industrial enterprises, social networks, hospitals, will be stored in a cloud in unorganized order. Now, we will apply a hierarchical clustering algorithm in the data center so that the data is sorted, such as educational data, medical data, as well as industrial data, etc. This will help end users to quickly retrieve data. The proposed model will also provide statistics on the effective use of cloud storage space, which is being employed by different types of organizations. This statistic will help to manage data in data centers.

CONCLUSION

Storage cloud is a promising technology that helps large organizations to store and manage their huge amount of data. Various kinds of work have been done in this area in order to increase the productivity of cloud computing, since one of the most important issues that should be considered in the cloud storage is quick access to the data stored in it. The approach we are proposing involves the implementation of a hierarchical clustering algorithm in the cloud-data centers for the organization of data according to their type. The proposed method has its advantages, since it provides fast access to data, statistics on the use of disk space in the cloud, shows scalability and helps in the analysis of large amounts of data that are inhomogeneous in nature. Future work on the proposed model is the application of other clustering algorithms in cloud storage systems and serve to compare the results to find the best clustering algorithm for cloud storage systems.

REFERENCES

1. **Boyko N. 2016** Application of mathematical models for improvement of “cloud” data processes organization / N. Boyko // Mathematical Modeling and Computing : scientific journal "Computational problems elektotehniky" – Vol. 3 No. 2 – Founder and Publisher Lviv Polytechnic National University. – P. 111-119.
2. **Boyko N. 2017** Use of a cloud storage for implementation informational proce / N. Boyko // ECONTECHMOD : an international quarterly journal on economics of technology and modelling processes – Vol. 2 No.6. – Branch in Lublin: Polish Academy of Sciences. – P. 3-8.
3. **Boyko N. 2016** A look trough methods of intellectual data analysis and their application in informational systems / Nataliya Boyko // Computer Science and Information Technologies CSIT 2016: Materials of the XI International Scientific and Technical Conference CSIT 2016. - Lviv: Lviv Polytechnic Publishing House - P. 183-185.
4. **Iudin O. K. 2013** Cloud technologies of organization of information corporate networks / O.K. Yudin, R.V. Ziubina, T.V. Ziubin // Informatsiina bezpeka. - T. 11. - No. 3. - P. 112-127

5. **.Boyko N. 2017** Building computer vision systems using machine learning algorithms / N. Boyko, N. Sokil // ECONTECHMOD : an international quarterly journal on economics of technology and modelling processes – 2017. Vol. 2 No.6. – Branch in Lublin: Polish Academy of Sciences. – P. 15-20**.**
6. BSA Global Cloud Computing Scorecard [Electronic resource]. — Access mode: http://cloudscorecard.bsa.org/2013/assets/PDFs/ BSA_GlobalCloudScorecard2013.pdf (in Ukrainian)
7. **Kroes N. 2011** Towards a European Cloud Computing Strategy / Neelie Kroes [Electronic resource]. — Access mode: http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/11/50 (in Ukrainian)
8. **Fedorov A. 2012** Golden clouds [Electronic resource] / A. Fedorov // IT-manager. - Access mode: http://www.it-world.ru/download/pdf/itmanager/100/ITM_02-2012_end.pdf (in Russian).
9. Cloud technology. Microsoft for educational institutions 2017 [Electronic resource]. - Access mode: http://pro-spo.ru/cloud-technology/3295-external-technologies-microsoft-for educational institutions (in Russian).
10. **Seidametova Z.S. 2012** Cloud services in education [Electronic resource] / Z.S. Seidametova, S.N. Seitvelyeva. - Access mode: http: //www.smartlearn.ru/site-admin/posts/563-external -services-building.IBM Smart Cloudfor Education. – Access mode: http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:IBM_SmartCloud_for_Education (in Russian)
11. **Raman T. V. 2008** Cloud computing and equal access for all. Proceedings of the 2008 international cross disciplinary conference on Web accessibility (W4A), Apr.2008, 1-4. DOI= http://doi.acm.org/10.1145/1368044.1368046.
12. **Foster I**. **2008** Cloud computing and grid computing 360-degree compared / I. Foster, Y. Zhao, I. Raicu, S. Lu // Grid Computing Environments Workshop, 2008, GCE'08., P. 1-10.
13. **Hayes B. 2008** Cloud computing. Communications of the ACM, vol. 51, no. 7. – P. 9-11.
14. **Mladen A. 2008** Cloud computing - Issues, research and implementations. Proceedings of the International Conference on Information Technology Interfaces (ITI’08). - P. 31-40.
15. **Philip K. McKinley, Farshad A. Samimi, Jonathan K. Shapiro, and Chiping Tang**. **2006** Service Clouds: A Distributed Infrastructure for Constructing Autonomic Communication Services. Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06). – P. 341-348.
16. **Robert L. Grossman, Yunhong Gu, Michael Sabala, and Wanzhi Zhang. 2010** Compute and storage clouds using wide area high performance networks. Future Generation Computer Systems, vol. 25, no. 2. P. 179-183.
17. **Milojicic D. 2015** Cloud computing: Interview with Russ Daniels and Franco Travostino. IEEE Internet Computing, vol. 12, no. 5. – P. 7-9.
18. **Nivikov PS 1973** Elements of mathematical logic. 2 ed. / P. S. Novikov. - Moscow: Science. - 400 p.
19. **Gilbert D. 1947** Fundamentals of Theoretical Logic / Gilbert D., V. Akkeman. - Moscow: GIIL – 302 p.
20. **Elkan C. 2003** Using the triangle inequality to accelebrate k-means / C. Elkan // In Proceedings of the Twelfth International Conference on Machine Learning – pp. 147–153.