Influence of Vantage Point-Based Dervied Genetic Key Matching For Heart Disease Diagnosis From Multiple Grid Location

K. Ashokkumar[1] and C. Chandrasekar[2]

Department of Computer Science and Engineering, Sathyabama University, Chennai, India.
Department of Computer Science Periyar University, Chennai, India.

(Received: 11 August 2015; accepted: 09 September 2015)

With huge quantity of data distributed in scientific and industrial world, demand for parallel accessing of information motivates the usage of Grid. Grid computing is an emerging infrastructure modeling for broad variety of disciplines with high volume and varied data sets. Due to the limitation of parallel access of data in the grid as well as satisfaction for both service provider and client related requirements, efficient data integration becomes an important challenge. In this paper, an efficient technique called, Derived Genetic Key Matching (DGKM) is developed for quick parallel accessing of data for heart disease diagnosis from multiple grid location and seamless data integration spread over the disturbed grid servers is introduced. Synchronization of storage key to the grid location is done to identify the request data based on the factors leading to heart disease for multiple users (i.e. patients) at different location and therefore improving the data integrity rate. DGKM in distributed grid services allows for parallel and integrated data accessing (i.e. accessing different features) with derived gene populations of key matching indexes, aiming at reducing the time taken for key matching. Finally, the Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm is developed to optimize different data grid storage with the objective of returning the result based on the factors resulting in heart disease to corresponding grid server location, aiming at improving the speed of parallel data accessing from distributed grids. The proposed technique is implemented by GridSim, a resource modeling and application scheduling for parallel computing. DGKM performance are tested with grid file accessing for online data repositories using Cleveland Clinic Foundation Heart disease data set available from UCI repository with metrics such as data grid access speed, data integrity rate, time taken for key matching, and accuracy of grid location identification. Experiment results show that the proposed technique achieve better performance by improving the data access speed by 17.04% and accuracy of grid location identification by 15.13% compared to state-of-the-art works.

Key words: Grid computing, Derived Genetic, Key Matching, Parallel accessing, Gene populations, Parallel computing.

The Grid is a combination of hardware and software infrastructure providing reliable, dependable, pervasive, and economical access with high-end computational capacity for broad disciplines with high volume and varied data sets. The data grid should be handled efficiently in a parallel manner at different location to improve the computational and resource efficiency of the Grid Data Environment.

In[1], a dynamic data replication algorithm was introduced with the objective of reducing the job execution time with effective network usage. The dynamic data replication algorithm minimized the data access time by improving the overall performance of the system. Enhanced Dynamic Hierarchical Replication[2] also minimized the data access time with the introduction of Weighted Scheduling Strategy (WSS). Virtualization of process for big data in cloud environment[3] was handled using network related computational resources. A review of grid allocation was

* To whom all correspondence should be addressed.
E-mail: ashokkumar.cse@sathyabamauniversity.ac.in

performed in[4].

Nash Equilibrium was introduced in[5] with the objective of improving the computational capability while assigning grid by several users. However, in many real scenarios, involving multilingual solutions, accuracy and time for allocation of grid has to be suggested. Therefore, key matching indexes based on gene population and Vantage Point is presented in this paper.

Allocation of grid in cloud computing becomes more and more complicated with the applications and characteristics of multi server system. In[6], a framework for optimal multi server configuration was introduced with the objective of optimizing the server speed without compromising the quality of service. An adaptive algorithm was introduced in[7] to ensure parallel processing by reducing the prediction errors in a significant manner. In[8], multi objective game theory was introduced to solve the problem of fairness and efficiency using communication and storage aware multi objective algorithm.

Heterogeneous allocation of resources was performed in[9] by applying K-Means clustering algorithm with the objective of minimizing the workload requirements. Efficient data storage and parallel processing in mobile cloud was handled in[10] using an approach called k-out-of-n computing. However data integrity rate lack in the above paper which is addressed through synchronization of storage key with grid location in DGKM technique.

Tremendous progress has been made in the recent past years for designing and developing incredible parallel computing architectures. In[11], parallel computation was performed in an efficient manner using Exchanged Cross Cube (ECC) technique. Fuzzy logic for composition optimization[12] was introduced with the objective of assigning weights across different geographical locations (grid). An ontology based approach was also designed for efficient allocation of grid.

Attribute based solution was introduced in[13] to address scalability with respect to grid and provided fine grained access control in cloud environment. Also computational complexity was reduced by applying Attribute Based Encryption (ABE). In order to perform parallel processing and improve data sharing in cloud, object centered approach was introduced in[14] ensuring accountability. Continuous aggregation of queries to ensure parallel processing[15] used cost-based query planning was introduced. However, the time taken for allocation of grid remained unaddressed. Therefore, to reduce the time, key matching indexes based on derived gene population is introduced in the DGKM technique.

A hybrid approach was introduced in[16] to improve the rate of record matching using privacy preserving partitioning. Another method called Saturn [17] was introduced to reduce the load balancing and minimize the rate of fault using horizontal and vertical replication. In[18], fault tolerance with distributed systems was introduced to address issues related to allocation of resources using Byzantine fault tolerance mechanism. Fast allocation of resources using keywords search was performed in[19] to reduce the query response time in a significant manner. Extended Sub tree[20] improved the runtime efficiency by applying distance function. Classification of classical swine flu virus[21] developed potential for new vaccines using antigenic characterization. In[22], a review for treatment of pilon fracture was presented.

In this paper with the input data drawn from Cleveland Clinic Foundation Heart disease data set from UCI repository, three strategies are proposed, first a novel synchronization technique for storage key to grid location that takes into account input attributes (i.e., with the dataset description from table 1) and grid location attributes is presented. Second a novel matching indexes strategy, called Key Matching Indexes based on Derived Gene Population is presented. The Key Matching Indexes based on Derived Gene Population improves the proposed technique by using a fitness function that evaluates the solution domain (i.e. the resultant factors leading to hear disease) in an efficient manner. Finally, a Vantage Point Tree Index improves the data integrity speed rate in an extensive manner.

The rest of this paper is organized as follows: Section 2 proposes our Derived Genetic Key Matching (DGKM) technique for heart disease diagnosis. In Section 3, experimental settings for DGKM technique is presented with the help of the dataset description provided in table 1. In Section 4 the discussion with the table values and graph form is presented. Finally, Section 5 concludes our work.

MATERIALS AND METHODS

Derived Genetic Key Matching

The Grid is a combination of hardware and software infrastructure that offers reliable, dependable, pervasive, and economical access to high-end computational capacity. Parallel accessing of data grid at different location needs to be handled properly, to improve the computational and resource efficiency of the Grid Data Environment.

In this section, the new technique called Derived Genetic Key Matching for quick parallel accessing is presented. For improving the grid location identification, reducing the time taken for key matching and improve data integrity rate, it is better to access data from multiple grid location and perform synchronization improving the data access speed. This work presents a technique using derived genetic key matching for heart disease diagnosis. The architecture of the proposed Derived Genetic Key Matching (DGKM) technique is shown in Figure 1.

Figure 1 given above shows the Block diagram of DGKM technique with the input obtained from Heart disease data set. As sown in the figure, the DGKM technique is divided into three parts. The first part performs efficient synchronization of storage key to grid location for efficient heart disease diagnosis aiming at improving the data integrity rate. Key matching indexes using derived gene population is performed that introduces the features and factors resulting in heart disease in the second part with the aim of reducing the time taken for key matching. Finally, Vantage Point Tree index using Berkeley Key Matching algorithm results in the optimized data grid storage and therefore improving the data grid access rate. The elaborate description of DGKM technique is discussed in the forthcoming sections.

Design of synchronization of storage key with grid location

The first step in the design of Derived Genetic Key Matching (DGKM) technique is the efficienct synchronization of storage key to the grid location. The synchronization is performed to identify the request data (i.e. factors causing heart disease) for multiple users (i.e. patients) from different location and therefore improving the data integrity rate.

Figure 2 shows the block diagram of synchronization process that involves efficient synchronization of storage key to grid location. As shown in the figure, given an input with multiple users '$User_i = User_1, User_2, \dots, User_n$' and '$SK$' storage keys, with grid locations $GridLoc_i = GridLoc_1, GridLoc_2, \dots, GridLoc_n$' synchronization of storage key to grid location involves associating storage key to grid location. Multiple users located at different places request for the factors leading to heart disease.

A string of features '$SK_1, SK_2, \dots, SK_n$' is used to represent each storage key. The mathematical formulation for each feature is as given below

$$SK_i = (Key_{id}, L_i) \qquad \text{...(1)}$$

From (1), the storage key 'SK_i' includes two attributes, where 'Key_{id}' represents the key allotted for each user '$User_i$' (i.e. patient) and 'L_i'

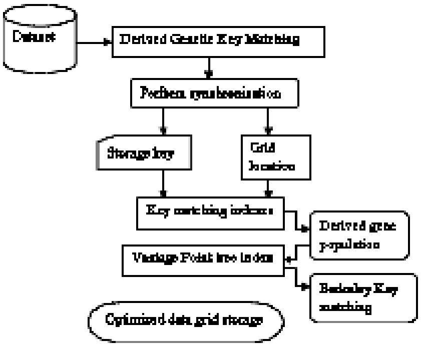


Fig. 1. Block diagram of DGKM technique

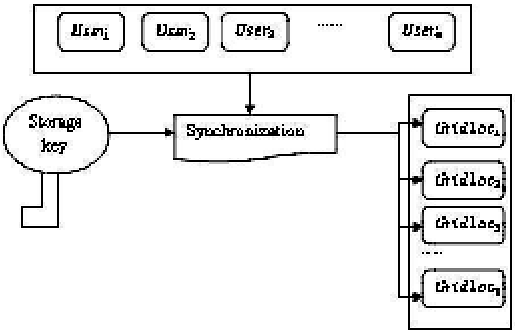


Fig. 2. Block diagram of synchronization of storage key with grid location

denotes the length of the storage key respectively. The synchronization of storage key to the grid location problem now minimizes to multiple key indexing matching. Let us consider the input user ‘’ with ‘’ where ‘’ is the ‘’ including attributes as given below

$$Input\ Attr\ (SK_i) = (Key_{id}\ (User_k), L_i\ (User_k)) \qquad ...(2)$$

Let us consider the grid locators as ‘$GridLoc_1, GridLoc_2, \ldots\ldots, GridLoc_n$’ where ‘$GridLoc_i = GridLoc_{i,1}, GridLoc_{i,2}, \ldots\ldots, GridLoc_{i,j}$’. The grid locators include the information relating to the factors resulting in heart disease. For example, a patient with fasting blood sugar > 120 mg/d with maximum heart rate achieved with exercise induced angina has the highest probability of obtaining heart disease. Here, ‘$GridLoc_{i,k}$’ is the ‘kth’ input feature of ‘’ location comprising of attributes as given below

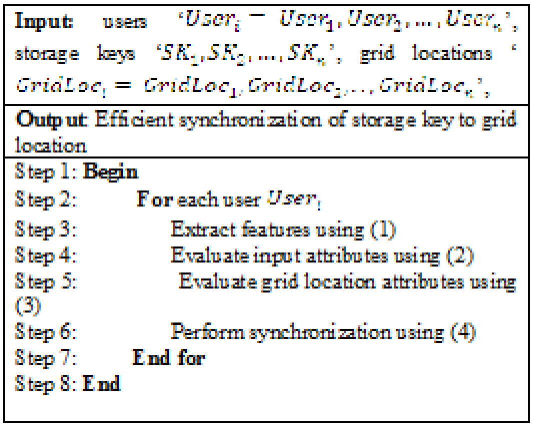
$$Grid\ Location\ Attributes = Key_{id}\ (GridLoc_{i,k}), L_i\ (GridLoc_{i,k})) \qquad ...(3)$$

From (2) and (3), after each iteration ‘$I = I_1, I_2, \ldots, I_n$’ corresponds to mapping function ‘’ from input attributes ‘$Input\ Attr$’ (i.e. 13 attributes provided in table 1) to grid location attributes ‘$Grid\ Location\ Attributes$’ such that the following condition is met.

$$I_m = f\ (User_m) = GridLoc_{i,m} \qquad ...(4)$$

From (4), synchronization of storage key to the grid location is performed in an efficient manner to identify the request data for multiple users (i.e. patients) at different location (i.e. from different hospitals across the region) in a significant manner. This in turn improves the data integrity rate. The Storage Key Grid Location Synchronization (SKGLS) algorithm works as given below.

The SKGLS algorithm given above (Figure 3) performs efficient synchronization of storage key to grid location with the objective of improving the data integrity rate. For each user, the algorithm starts with the extraction of possible features. With the evaluated features, input attributes and grid location attributes are obtained. With the aid of these two attributes, synchronization is performed in an efficient manner.



```
Input: users '$User_i = User_1, User_2, \ldots, User_n$', storage keys '$SK_1, SK_2, \ldots, SK_n$', grid locations '$GridLoc_i = GridLoc_1, GridLoc_2, \ldots, GridLoc_n$',
Output: Efficient synchronization of storage key to grid location
Step 1: Begin
Step 2:     For each user $User_i$
Step 3:         Extract features using (1)
Step 4:         Evaluate input attributes using (2)
Step 5:         Evaluate grid location attributes using (3)
Step 6:         Perform synchronization using (4)
Step 7:     End for
Step 8: End
```

Fig. 3. Algorithm for SKGLS

Construction of Key Matching Indexes based on Derived Gene Populations

The second step in the design of Derived Genetic Key Matching (DGKM) technique in distributed grid services allows for parallel and integrated data accessing with derived gene populations of key matching indexes, aiming at reducing the time taken for key matching. In key matching indexes based on derived gene population, efficient matching is performed based on the dataset description from table 1. Efficient key matching indexes is handled using derived gene population with the aid of fitness function to evaluate the solution domain (i.e. key matching indexes).

As shown in figure 4, the DGKM technique in distributed grid services where the patients are located in different regions allows for parallel and integrated data accessing regarding the factors that influence heart disease using heart disease dataset with derived gene populations (i.e. ‘$User_1, User_2, \ldots, User_n$’) of key matching indexes. Efficient key matching indices is performed

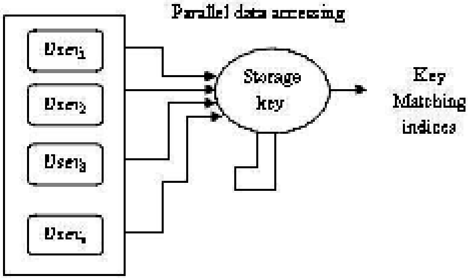


Fig. 4. Block diagram of Key Matching Indexes

using fitness function with derived gene population (i.e. different users) as given below

$$Fitness = \{User_i \,| f(User_i) = SK_i, GridLoc_{i,m}\} + \{f(User_i) = SK_j, GridLoc_{j,n}\} > \alpha \quad ...(5)$$

From (5), efficient key matching indices 'KMI' is performed where the first statement marks the storage keys assigned to different users. Each storage key includes the list of attributes provided in table 1 and the matching factors that result in heart disease. The second statement marks the features (i.e. users) with storage key not matched by function 'f'. The key matching indexes using Pearson Correlation Coefficient is as given below

$$KMI = \left(\frac{\sum_{i=1}^{n} User_i . Key_{id}}{\sum_{i=1}^{n} User_i} \right) \quad ...(6)$$

From (6), the key matching indexing '' is evaluated on the basis of similarity between derived gene population (i.e. patients) '' and storage keys '' respectively, where each storage key represents the attributes with the description. For example, the attributes with the description include 'number of major vessels covered by flouoropsy = 0/1/2/3' and so on as listed in table 1. So the storage key that matches with that of the users are only subjected to key matching which in turn reduces the time taken for key matching.

Vantage Point (VP) Tree Indexed Berkeley Key matching

In this section, the application of Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm is explained for efficient heart disease diagnosis. The Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm is developed to optimize different data grid storage to corresponding grid server location. Vantage Points are used to split the grid storage that include the resultant factors leading to heart disease into tree structure and perform the process in a hierarchical manner. VP Tree Indexed Berkeley key matching algorithm in DGKM improves the speed of parallel data accessing from distributed grids that includes the cause for heart disease from different regions. Tree indexed key matching process is periodically done for multiple requests obtained from different patients from different locations for data accessing.

Let us discuss the Vantage Point (VP) Tree to partition the grid storage around selected Vantage Points at several levels to form a hierarchical tree structure. The Tree Indexed Berkeley Key algorithm is then used for effective optimization of grid storage to corresponding grid server location. Vantage Point (VP) Tree consists of a binary tree where the internal node is formulated as given below

$$Node_i = (VP, M_{dis}, R_{leaf}, L_{leaf}) \quad ...(7)$$

From (7) the internal node 'Node' consist of vantage point 'VP', midpoint distance among all the distances of vantage points represented by 'M_{dis}' and pointers to left and right leaf represented by 'L_{leaf}' and 'R_{leaf}' respectively.

Left leaf of the node indexes the vantage points whose midpoint distances from 'VP' are less than or equal to 'M_{dis}', and right leaf of the node indexes the points whose distances from 'VP' are greater than or equal to 'M_{dis}'. In Vantage Point (VP) Tree Indexed form, instead of pointers to the left and right leaf, references to the data Vantage Points are used.

For each user 'User' in every internal node, the midpoint 'M' is used for partition with respect to the first vantage point 'VP_1', and medians, 'M_1' and 'M_2' are used in reference with respect to the second vantage point 'VP_2'. With the aid of heart disease dataset, each user forms the patients with the first vantage point for example representing the fasting blood sugar (fbs) > 80, second vantage point being fasting blood sugar > 100 and so on.

Figure 5 given above shows the block diagram of Vantage Point (VP) Tree Indexed Berkeley Key where the leaf node consists of the distances between data points in and vantage points of that leaf are used. The distances between

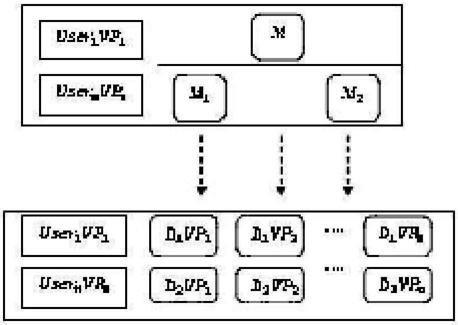
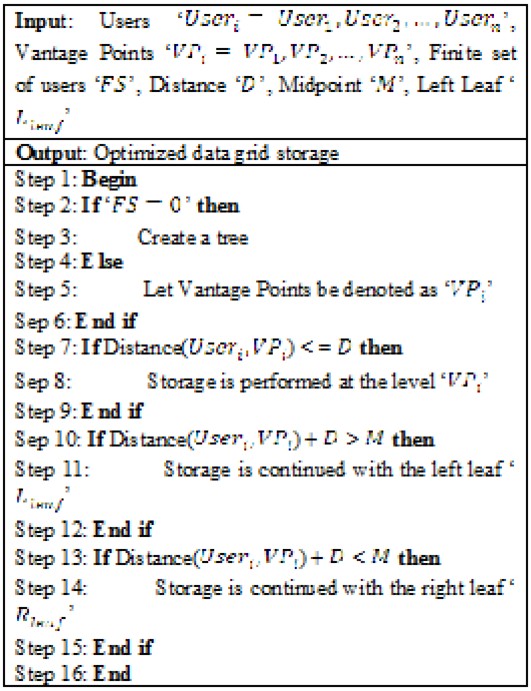


Fig. 5. Block diagram of Vantage Point (VP) Tree Index

data points using heart disease dataset represent the various classifications and factors resulting in heart disease. From the figure 5, 'D_1VP_1' and 'D_2VP_1', are the distances between the first and second vantage points respectively, where in our heart disease diagnosis example, the first vantage point being fbs > 80, second vantage point being fbs > 100. The algorithmic description of VP Tree Indexed Berkeley key matching is given below in figure.

Figure 6 given above shows the algorithmic description of VP Tree Indexed Berkeley key matching. Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm is designed as shown above with the objective of optimizing several data grid storage with their respective grid server location.



```
Input: Users '$User_i = User_1, User_2, ..., User_n$', Vantage Points '$VP_i = VP_1, VP_2, ..., VP_n$', Finite set of users '$FS$', Distance '$D$', Midpoint '$M$', Left Leaf '$L_{leaf}$'
Output: Optimized data grid storage
Step 1: Begin
Step 2: If '$FS = 0$' then
Step 3:         Create a tree
Step 4: Else
Step 5:         Let Vantage Points be denoted as '$VP_i$'
Sep 6: End if
Step 7: If Distance($User_i, VP_i$) $<=$ $D$ then
Sep 8:          Storage is performed at the level '$VP_i$'
Step 9: End if
Sep 10: If Distance($User_i, VP_i$) $+ D > M$ then
Step 11:        Storage is continued with the left leaf '$L_{leaf}$'
Step 12: End if
Step 13: If Distance($User_i, VP_i$) $+ D < M$ then
Step 14:        Storage is continued with the right leaf '$R_{leaf}$'
Step 15: End if
Step 16: End
```

Fig. 6. VP Tree Indexed Berkeley key matching algorithm

The motivation behind the application of VP Tree Indexed Berkeley key matching algorithm in DGKM using heart disease dataset is to further enhance the parallel data accessing speed from distributed grids. As shown in figure, the tree indexed key matching process is continuously performed in an iterative manner for multiple user requests of data accessing. The performance of DGKM technique is tested with grid file accessing for online data repositories.

Experimental settings

To quick parallel accessing of data from multiple grid location using Derived Genetic Key Matching (DGKM) technique, experiments were conducted based on Gridsim simulator and applied in JAVA using Cleveland Heart disease dataset from UCI repository. Cleveland Heart disease dataset consists of 76 attributes, whereas the experiments are conducted with the aid of 14 of them. Experiments with the Cleveland database using DGKM technique concentrated on attempting to differentiate between the presence of disease by the values 1, 2, 3, 4 and the absence of disease by the value 0. The dataset description of Cleveland Heart disease dataset from UCI repository is provided in table 1.

The work Derived Genetic Key Matching (DGKM) technique is compared against the existing Prefetching based Dynamic Data Replication Algorithm (PDDRA) [1] and Enhanced Dynamic Hierarchical Replication in Data Grid (EDHR-DG) [2] technique. The experiment is conducted on the factors such as data grid access speed, data integrity rate, time taken for key matching, and accuracy of grid location identification.

Table 1. Dataset description

1	Age	Numerical
2	Sex	Male, Female
3	Chest Pain Type	1, 2, 3, 4
4	Resting Blood Pressure	Numerical
5	Serum Cholestoral in mg/dl	numerical
6	Fasting Blood Sugar > 120 mg/d	Yes, No
7	Resting Electrocardiograph results	0, 1, 2, 3
8	Maximum heart rate achieve	Numerical
9	Exercise induced angina	Yes, No
10	ST depression induced by exercise relative to rest	Numerical
11	Slope of peak exercise	Numerical
12	Number of major vessels colored by flourosopy	0 – 3
13	Thal	Normal, Fixed defect, reversible defect
14	Disease diagnosis	Presence (1), Absence (0)

DISCUSSION

The result analysis of Derived Genetic Key Matching (DGKM) technique is compared with the existing Prefetching based Dynamic Data Replication Algorithm (PDDRA) [1] and Enhanced Dynamic Hierarchical Replication in Data Grid (EDHR-DG) [2] technique. Table 2 represents the data integrity using JAVA platform and comparison is made with two other methods, namely PDDRA [1] and EDHR-DG [2].

Impact of data integrity rate

Data integrity rate is ensured by checking whether the data is recorded exactly as intended by the patient registered and queried from different location and if so measures the changes being made. The mathematical formulation of data integrity rate is given as below

$$DIR = \sum_{i=1}^{n} (User_i * Data_{size}) - (Data_{dropped}) \quad \text{...(8)}$$

From (8), the data integrity rate 'DIR' is measured using the actual data size '$Data_{size}$' and the data dropped '$Data_{dropped}$' with respect to the number of users '$User_i$'.

Figure 7 shows the result of data integrity rate efficiency that measures the amount of data successfully sent on the basis of the drop rate versus the varying number of data send in the range of 50 – 350 KB. To better perceive the efficacy of the proposed DGKM technique, substantial experimental results are illustrated in Figure 7 and compared against the existing PDDRA [1] and EDHR-DG [2] respectively.

The data integrity rate for different observations made by several users (i.e. patients) is performed at different time interval is shown above. Higher, the number of data being sent (i.e., the observation from different patients located at different regions), more successful the technique is. The results reported here confirm that with the increase in the number of data being sent, the data integrity rate efficiency also increases. The process is repeated for 7 different data packets.

Table 2. Tabulation for data integrity rate

Data size (KB)	Data integrity rate (KB) DGKM	PDDRA	EDHR-DG
50	231	224	218
100	285	286	276
150	320	300	290
200	300	280	270
250	325	305	285
300	315	295	285
350	345	325	315

As illustrated in Figure 7, the proposed DGKM technique performs relatively well when compared to two other methods PDDRA [1] and EDHR-DG [2]. The data integrity rate efficiency using DGKM technique is improved with the synchronization of storage key with the grid location where data transmission is performed in and out of its neighbors. As a result, by applying mapping function for each user based on the storage key, results in the improvement of data integrity rate efficiency using DGKM technique by 3.5% and 7.72% compared to PDDRA [1] and EDHR-DG [2] respectively.

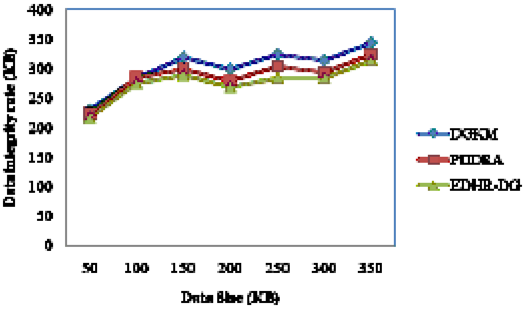


Fig. 7. Measure of data integrity rate

Impact of time taken for key matching

The time taken for key matching based on the training and test dataset is the amount of time taken to match the key with respect to the total number of user (i.e. patient) who are ready for acquiring the information regarding the factors resulting in heart disease. It is mathematically formulate as given below.

$$Time_{keymatc\ hing} = Time\ (single\ user) * n \quad \text{...(9)}$$

From (9), the time for key matching '$Time_{keymatching}$' is obtained using the time for single user '$Time\ (single\ user)$' and total number of users 'n'.

In table 3 we further compare the time taken for key matching with different number of users (i.e. for different number of patients). The experiments were conducted with 70 observations (i.e., users) and the time taken for key matching obtained is measured in terms of milliseconds (ms).

Table 3. Tabulation for time taken for key matching

No. of users (n)	Time taken for key matching (ms) DGKM	PDDRA	EDHR-DG
10	3.91	4.72	5.42
20	5.23	9.26	9.25
30	8.54	12.57	14.59
40	7.22	11.25	13.27
50	11.35	15.38	20.40
60	10.82	14.85	19.87
70	15.23	19.26	24.28

In order to reduce the time taken for key matching for further transfer of data packets, the time taken for single user is considered. In the experimental setup, the number of user ranges from 10 to 70 is illustrated in figure 6. The time taken for key matching using the technique DGKM provides comparable values than the state-of-the-art methods.

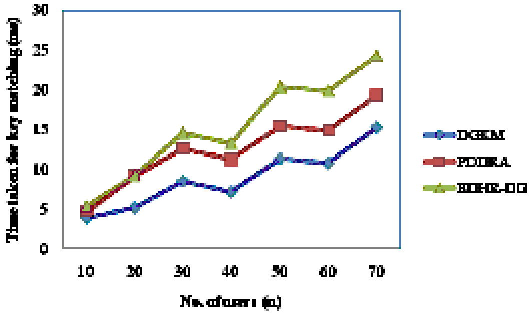


Fig. 8. Measure of time taken for key matching

The targeting results of number of users to measure the time taken for key matching using DGKM is compared with two state-of-the-art methods PDDRA and EDHR-DG in figure 8 is presented. Our technique DGKM differs from the PDDRA [1] and EDHR-DG [2] in that we have incorporated Key Matching indexes based on derived gene population and ensures parallel and integrated data accessing. Based on the fitness function, the results are generated and evaluate the solution domain in an efficient manner. As a result, the time taken for key matching is reduced by 48.88% and 61.12% compared to the PDDRA [1] and EDHR-DG [2] respectively.

Impact of data grid access speed

The data grid access speed is the rate at which the user is allocated with the grid based on the storage key. Higher the data grid access speed, more efficient the method is said to be and is measured in terms of kilo bits per second (KB/s). Table 4 shows the data grid access speed with respect to user and grid location with respect to seventy users.

Figure 9 given below shows the data grid access speed for DGKM technique, PDDRA [1] and EDHR-DG [2] versus seventy different users (i.e., vehicles). The data grid access speed rate returned over DGKM technique increases gradually for differing number of users.

Table 4. Tabulation for data grid access speed

Methods	Data grid access speed (KB/s)
DGKM	125
PDDRA	115
EDHR-DG	85

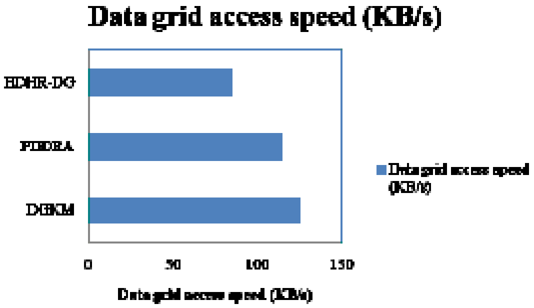


Fig. 9. Measure of data grid access speed

From figure 9, it is illustrative that the data grid access speed is improved using the proposed technique DGKM. This is because with the application of Vantage Point Tree Index, the data grid access speed is increased. With the help of Vantage Point Tree Index, to optimize different grid storage, the DGKM technique applies Berkeley key matching algorithm that helps in identifying the data drop rate at an early stage in an extensive manner. This in turn helps in improving the data access speed by 8% compared to PDDRA. In addition, by applying Berkeley key matching algorithm helps in improving the probability of successful data transmission using binary tree and therefore improving the data grid access speed by 26.08% compared to EDHR-DG.

Impact of accuracy of grid location identification

The accuracy of grid location identification is measured on the basis of the number of users who were successfully located with the grid and those who have accessed the information and the factors relating to heart disease. The accuracy is formulated as given below and is measured in terms of percentage (%).

$$A = \left(\frac{GridLoc_s}{n}\right) * 100 \qquad ...(10)$$

From (10), the accuracy of grid location identification 'A' is measured using the users who were successfully located with the grid '$GridLoc_s$' to the number of users 'n'. Higher, the accuracy of grid location identification, more efficient the method is said to be.

Table 5 and figure 10 shows the measure of accuracy of grid location identification. From the figure it is illustrative that the accuracy of grid location identification is improved with the increase in the number of users. However, comparatively betterment is observed using DGKM technique than PDDRA and EDHR-DG. This is because of the application of Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm that optimizes several data grid storage with their respective grid server location. This in turn improves the accuracy of grid location identification by 10% compared to PDDRA and 20.26% compared to EDHR-DG respectively.

Table 5. Tabulation for accuracy of grid location identification

No. of	Accuracy of grid location identification (%)		
users (n)	DGKM	PDDRA	EDHR-DG
10	70.23	60.35	50.14
20	74.32	69.29	64.23
30	78.15	73.12	68.06
40	76.23	71.20	66.14
50	79.49	74.46	69.40
60	78.12	73.09	68.03
70	84.55	79.52	74.46

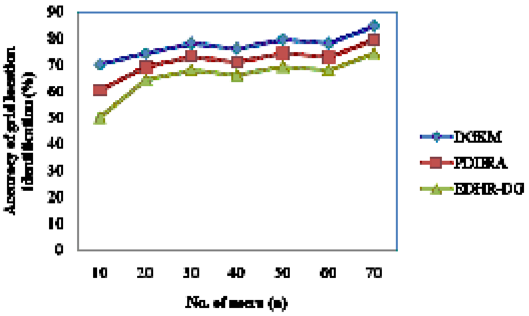


Fig. 10. Measure of accuracy of grid location identification

CONCLUSION

Derived Genetic Key Matching (DGKM) technique performs efficient parallel processing to reduce the time taken for key matching and improve data integrity rate for users on dynamic observations for heart disease diagnosis from various regions. We then showed how this technique can be extended to incorporate synchronization of storage key (i.e. the factors resulting in heart disease) to the grid location where the key factors are stored to improve the data integrity rate based on input attributes and grid location attributes. The synchronization of storage key to the grid location also provided efficient mapping for multiple users at different location based on the storage key and grid location and hence improved the data integrity rate. Next, the introduced key matching indexes based on derived gene population reduces the time taken for key matching in an efficient manner for parallel and integrated data accessing. This parallel and integrated data accessing for different patients relating to heart disease from different regions improves the accuracy of grid location identification. Finally, the Vantage Point (VP) Tree Indexed Berkeley Key matching algorithm improves the data grid access speed using Vantage Point Tree Index. In our experimental results the DGKM technique showed better performance than the PDDRA and ARM-RN and EDHR-DG over the parameters, data integrity rate, time taken for key matching, data grid access speed and accuracy of grid location identification. The results show that DGKM technique offers better performance with an improvement of data integrity rate by 5.61% and reduces the time taken for key matching by 55% compared to PDDRA and EDHR-DG respectively.

REFERENCES

1. Nazanin Saadat and Amir Masoud Rahmani, "PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids", *Elsevier,* 2012; **28**(4): 666 – 681.
2. Najme Mansouri and Gholam Hosein Dastghaibyfard, "Enhanced Dynamic Hierarchical Replication and Weighted Scheduling Strategy in Data Grid", *Elsevier,* 2013; **73**(4): 534 – 543.
3. Haiyan Guan, Jonathan Li, Liang Zhong, Yongtao Yu and Michael Chapman, "Process virtualization of large-scale lidar data in a cloud computing environment", *Elsevier,* 2013; **60**: 109 – 116.
4. Rajkumar Buyya, "Introduction to the IEEE Transactions on Cloud Computing", *IEEE Transactions on Cloud Computing,* 2013; **1**(1): 3 – 21.
5. Sudip Misra, Snigdha Das, Manas Khatua and Mohammad S. Obaidat, "QoS-Guaranteed Bandwidth Shifting and Redistribution in Mobile Cloud Environment", *IEEE Transactions on Cloud Computing,* 2014; **2**(2): 181 – 193.
6. Junwei Cao, Kai Hwang, Keqin Li and Albert Y. Zomaya, "Optimal Multi server Configuration for Profit Maximization in Cloud Computing", *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2013; **24**(6): 1087 – 1096.
7. Sheng Di, Cho-Li Wang and Franck Cappello, "Adaptive Algorithm for Minimizing Cloud Task Length with Prediction Errors", *IEEE Transactions on Cloud Computing,* 2014; **2**(2): 194 – 207.
8. Rubing Duan, Radu Prodan and Xiaorong Li, "Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds", *IEEE Transactions on Cloud Computing,* 2014; **2**(1): 29 – 42.
9. Qi Zhang, Mohamed Faten Zhani, Raouf Boutaba and Joseph L. Hellerstein, "Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud", *IEEE Transactions on Cloud Computing,* 2014; **2**(1): 14 – 28.
10. Chien-An Chen, Myounggyu Won, Radu Stoleru and Geoffrey G. Xie, "Energy Efficient Fault-Tolerant Data Storage and Processing in Mobile Cloud", *IEEE Transactions on Cloud Computing,* 2015; **3**(1): 28 – 41.
11. Keqiu Li, Yuanping Mu, Keqin Li and Geyong Min, "Exchanged Crossed Cube: A Novel Interconnection Network for Parallel Computation", *IEEE Transactions on Parallel and Distributed Systems (TPDS),* 2013; **24**(11): 2211 – 2219.
12. Amir Vahid Dastjerdi and Rajkumar Buyya, "Compatibility-Aware Cloud Service Composition under Fuzzy Preferences of Users", *IEEE Transactions on Cloud Computing,* 2014; **2**(1): 1 – 13.
13. Zhiguo Wan, Jun'e Liu and Robert H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing", *IEEE Transactions on Information Forensics and Security,* 2012; **7**(2): 743 – 754.
14. Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud", *IEEE Transactions on Dependable and Secure Computing,* **9**(4): 556 – 568.
15. Rajeev Gupta and Krithi Ramamritham, "Query Planning for Continuous Aggregation Queries over a Network of Data Aggregators", *IEEE Transactions on Knowledge and Data Engineering (TKDE),* 2012; **24**(6): 1065 – 1079.
16. Ali Inan, Murat Kantarcioglu, Gabriel Ghinita and Elisa Bertino, "A Hybrid Approach to Private Record Matching", *IEEE Transactions on Dependable and Secure Computing,* 2012; **9**(5): 684 – 698.
17. Theoni Pitoura, Nikos Ntarmos and Peter Triantafillou, "Saturn: Range Queries, Load Balancing and Fault Tolerance in DHT Data Systems", *IEEE Transactions on Knowledge and Data Engg (TKDE),* 2012; **24**(7): 1313 – 1327.
18. Bharath Balasubramanian and Vijay K. Garg, "Fault Tolerance in Distributed Systems using Fused Data Structures", *IEEE Transactions on Parallel and Distributed Systems (TPDS),* 2013; **24**(4):1 – 16.
19. Yufei Tao and Cheng Sheng, "Fast Nearest Neighbor Search with Keywords", *IEEE Transactions on Knowledge and Data Engineering (TKDE),* 2014; **26**(4): 1 – 13.
20. Ali Shahbazi and James Miller, "Extended Subtree: A New Similarity Function for Tree Structured Data", *IEEE Transactions on Knowledge and Data Engineering (TKDE),* 2014; **26**(4): 1 – 14.
21. Jin-Liang Wang, Quan-Wen Sun, Bing-Mei Dong, Feng Wei, Jin-long Chen, Yan-kai Dong,Ai-Hua Wang* and Zhi-Qiang Shen,"Antigenic Characterization of the Glycosylated E2 Proteins of Classical Swine Fever Virus", *Journal of Pure and Applied Microbiology,* **9**(2)
21. Shijuan Xie, Dan Jin1, Bin Yu, Yafei Xu and Zhenlyu Zou," Comparison between Staged ORIF and EFLIF in Treatment of Pilon Fracture: *A Systematic Review",* **9**(2)