Secure Data Management And Transaction of Heterogeneous Large Data Sets In Grid Computing Environment

K. Ashokkumar M.E, Dr. C. Chandra sekar Ph.D

Computer Science & Engineering Department, Sathyabama University.
Professor, Dept of Computer Science, Periyar University

Abstract -- Data management is one in every of the difficult problems perpetually in grid computing and its environments. As a result of grid computing systems and its applications deals with terribly giant information set, attributable to heterogeneous grid resources that happiness to totally different organizations and locations with different access policies. Providing security for grid is neither a simple nor a tough task. With this abundant interest, security becomes necessary to produce authentication, authorization, resource protection, secure communication, information confidentiality, information integrity, trust policies management, user key and document management, service protection, and network security. During this proposal we have a tendency to square measure planning to give security for the resources mistreatment thereforeme reliable security mechanisms so on preserve the heterogeneous information gift within the distributed systems. And conjointly focusing, however with efficiency integrate and method the info set from the heterogeneous atmosphere and also information integration strategies, that determine records belongs to a similar cluster of person who reside in multiple locations, square measure essential to those efforts.

Keyword: grid computing, data integartion, grid security, large data set, heterogeneous resources.

I. Introduction

A. Grid Computing

There has been a surge of interest in grid computing, a way to enlist large numbers of machines to work on multipart computational problems such as circuit analysis or mechanical design. There are excellent reasons for this attention among scientists, engineers, and business executives. Grid computing enables the use and pooling of computer and data resources to solve complex mathematical problems. The technique is the latest development in an evolution that earlier brought forth such advances as distributed computing, the worldwide web, and collaborative computing.

Herer We need one supervised system with enough memory and well computation power, but in this world no single system has been such kind of things as like our expectation. So we need to combine together few systems to make one cluster system. This cluster system can run the application with more size. But our problem is having all data set in different place. Cluster formation can happen with few systems only. And also we need more processing power and computation power. And can't expect such things from cluster system.

To avoid this problem, can go for grid system. We may call this as group of cluster can form a grid system. In this grid system ,have multiple clusters in each and every data set. So, execution in the form of searching, matching everything is possible in data set. Because, each dataset has their own cluster system, so execution is easy. Finally, the interdomain security solutions used for grids should be able to interoperate with, rather than replace, the varied intradomain access management technologies inevitably encountered in individual domains..[1] [2] [9].

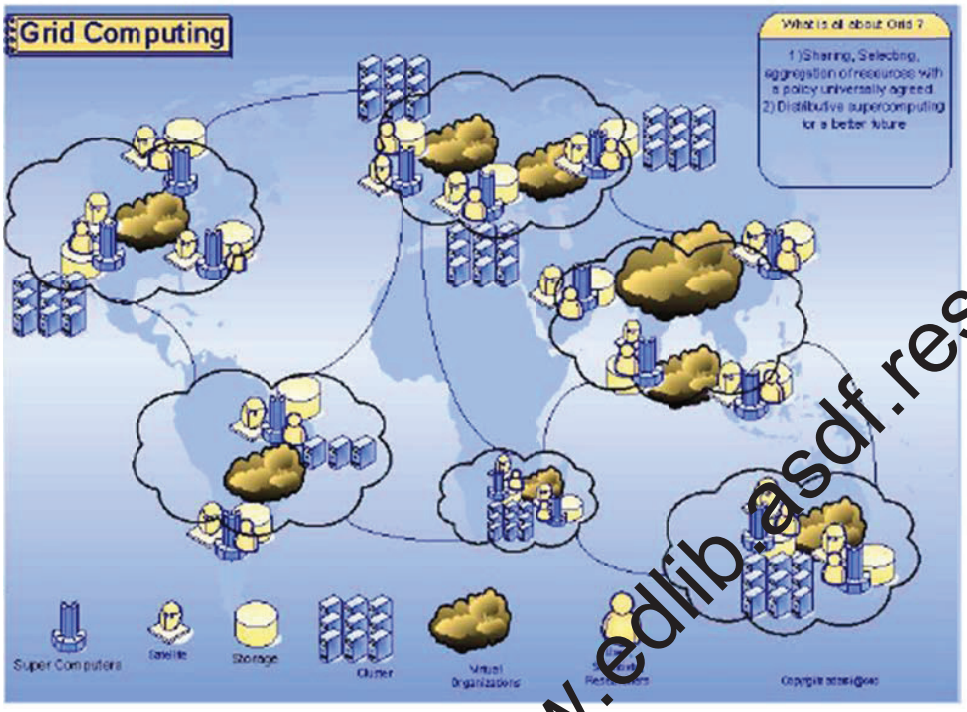

Fig.1 Grid computing [9]

II. Existing Information Retrieval Methods and Security in Grid Computing

Information Retrieval & Search Algorithms:

This provides the following characteristics:

1) **Accuracy:** The sheer volumes of information now stored in electronic media magnify deficiencies in recall (the percentage of relevant information retrieved. Giving the user reasonable-sized response with high precision can mean missing hundreds of relevant texts.
2) **Speed:** As the quantity of text that must be searched increases, the speed of searching can become a reserve bottleneck. In practical terms, the need for fast search means that more computational-intensive processing such as NLP techniques must either apply very selectively or run as "batch" indexing tool prior to retrieval.
3) **Consistency:** Many information retrieval environments require indexing of the text by the groups of indexers or by the authors. This leads to a decrease in accuracy from the inevitable inconsistencies, which automatic processing could help to avoid.
4) **Ease of use:** The growth of personal computer has made obsolete the traditional model of information retrieval with a trained human intermediary, giving systems responsiveness a high priority. [5]

A. Existing Method

There are several types of identifiers that can be used to link up with the upcoming enormous identification system known as India's Large dataset. For example – they can be as follows – driver license, ration card, election photo identity, PAN card, passport, health insurance card, bank account number, post office

account number, mobile phone number, landline phone number, email addresses. These shall be the criteria which the resident of the card may detail the authority with.[3] [13]

The working style of India's's large dataset card is based on 16 digit number. In which 12 digits only shown to public, and remaining 4 digits use for official purpose.

Upon the loss of the UID number, the card holder will have to go through a series of processes for the Identity Check and perhaps, since this process is a bit expensive, they will have to pay a small sum of money to the authority. *[a1 a2 a3 a4 a5 a6 a7 a8 a9 10 a11 a12]*

India's's Large dataset number schema will actually have not even 12 digits but only 11 digits. The last number will be the Implicit Version Number while the second, be the Check digit. So, that means that the large dataset number will only have 11 digits which really matters.

The numbers in UID will be non-repeating and non traceable or predictable and will be generated through computer algorithms. [13]

B. Access Method

The amount of accessing methods of those data sets is heavy, because we have to refer all the data set concurrently in order to get their information. Each and every data sets are located in different location. Accessing of those sets is not quite easy, and also can't expect uniqueness over there because each data sets have their own behavior.

Here we have to concentrate one thing primarily. That is data accuracy because user are getting information from different data set which are very huge in size, and also they have huge amount of entries. So when retrieve information from those data sets they have chance to get duplicate information like redundant data and modified data.

Second thing is accessing speed. We already know each data set is located in different location. Accessing of entire data set from single place is not easy. They need huge algorithms to implement this method.

C. Grid Security

We introduce the grid security downside with an example illustrated in Figure 1.1 . this instance, though somewhat contrived, captures necessary components of real applications.

III. Security Needs

Grid systems and applications could need any or all of the normal security functions, together with authentication, access control, integrity, privacy, and nonrepudiation. during this paper, we tend to focus totally on problems with authentication and access control. Speci cally, we tend to get to (1) give authentication solutions that enable a user, the processes that comprise a user\'s computation, and also the resources utilized by those processes, to verify every other\'s identity; and (2) enable native access management mechanisms to be applied while not modification, whenever doable. As are mentioned in Section four, authentication forms the inspiration of a security policy that permits numerous native security policies to be integrated into a global framework.

In developing a security design that meets these needs, we tend to additionally opt to satisfy the subsequent constraints derived from the characteristics of the grid surroundings and grid applications:

Single sign-on: A user ought to be ready to evidence once (e.g., once beginning a computation) and initiate computations that acquire resources, use resources, unharness resources, and communicate internally, while not additional authentication of the user.

Protection of credentials: User credentials (passwords, personal keys, etc.) should be protected.

Interoperability with native security solutions: whereas our security solutions could give interdomain access mechanisms, access to native resources can generally be determined by a neighborhood security policy that's enforced by a neighborhood security mechanism. it's impractical to change each native resource to accommodate interdomain access; instead, one or a lot of entities in an exceedingly domain (e.g., interdomain security servers) should act as agents of remote clients/users for native resources.

Exportability: we tend to need that the code be (a) marketable and (b) feasible in international testbeds. In short, theexportability problems mean that our security policy cannot directly or indirectly need the employment of bulk secret writing.

Uniform credentialscertication infrastructure: Interdomain access needs, at a minimum, a standard method of expressing the identity of a security principal like Associate in Nursing actual user or a resource. Hence, it's imperative to use a standard (such as X.509v3) for encryption credentials for security principals. Support for secure cluster communication. A computation will comprise variety of processes which will have to be compelled to coordinate their activities as a bunch. The composition of a method cluster will and can modification throughout the lifespan of a computation.

Hence, support is required for secure (in this context, authenticated) communication for dynamic teams. No current security resolution supports this feature; even MS-API has no provisions for cluster security contexts.

Support for multiple implementations: the protection policy shouldn\'t dictate a speci c implementation technology. Rather, it ought to be doable to implement the protection policy with a spread of security technologies, supported each public and shared key cryptography.

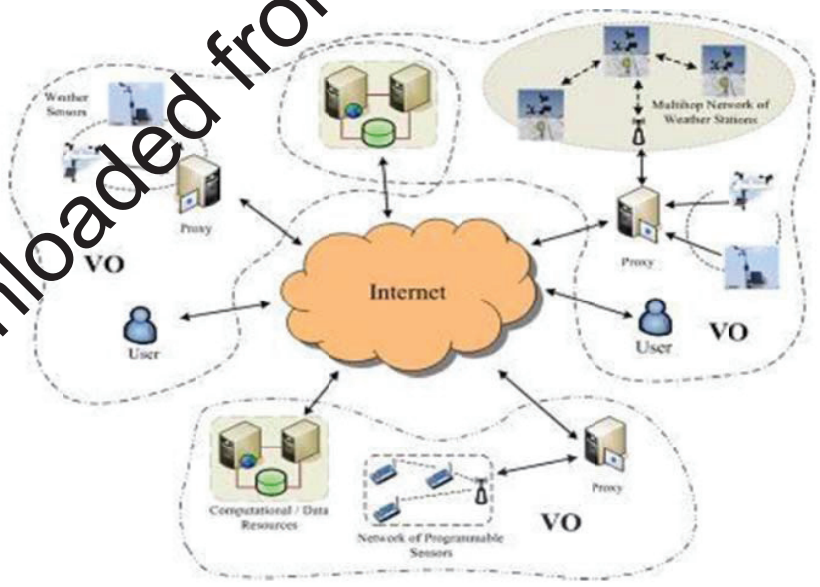

Fig.2 Grid Security

IV. Proposed Method

The amount of accessing methods of those data sets is heavy, because we have to refer all the data set concurrently in order to get their information. Each and every data set are located in different location. Accessing of those sets is not quite easy, and also we can't expect uniqueness over there because each data set have their own behavior. If we need all the data sets to be response quickly and perfectly means we have to change or modify those data set into similar style, like create or modify existing all data sets with more or less similar conditions, their working style, the way of accessing those sets, way of response of those data set should be same manner.

Here we create 13 digits id as large dataset card number.

[a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13]

- a1 – state a2 – district a3 -- taluk a4 –village
  a5 – street a6 -- door number a7 -- name of holder
  a8 -- date of birth of holder a9 –-gender a10 --license number a11 -- ration card number a12 -- pan card number a13--passport number

In this method passport dataset contains mobile number, landline number, E-mail address in their data set.

Our proposed work is, we have to assign id for each state, each district, each taluk, each village, and each street. Once we assigned id to those we can retrieve easily from them. By doing like this we have to design data sets in the form of [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10]

First nine fields should be same for all data sets. Next one field will change depends on their pattern. So we have to make entry of data in data set in above form. Above form is an example for license database. Here a10 indicates license number. How can we achieve this?

For that first when we enter the license holder's information in data set, that should follow the above manner like state, district, taluk, village, street, door no, holder's name, date of birth, and gender at last license card number. If we have license number like this, we can retrieve license number of holder from their license card data set very easy by matching first nine digit number of both large dataset card number and license card number.

Similarly in ration card data set a11 indicates ration card number. If we store ration card information in ration card data set like this , we can get information by comparing first nine digits number of both large dataset card and ration card. Because first nine digits is common for all data set, and next field only will get modify depends on their information. If its ration card data set then ration card number of particular holder information is there. If its pan card data set then a12th field consists of pan card number. Similarly in passport data set a 13^{th} field indicates passport number. We can do one thing here,When we make passport data set we can include holder's optional information like landline number, mobile number, E-mail address. When we retrieve passport information of particular holder it also retrieves mobile number, landline number, E-mail address.

At last our work is, We have to make all the information in data set should be in the form of digits, and have to assign number for each state, district, taluk, and village, street and also these are all act as primary key. What do you mean by primary key, it's a key attribute which is a combination of both unique value and not null value? So we assign state, district, taluk, village, street number as a primary key. Once we assigned we can retrieve those information by using query statement. By writing query for that we can retrieve information from those data set and will display easily.

"our method is just one time effort...once we did it we can use our system life long."

Efficient method: To find / locate the database

- Set id length of 13 bit id or more for database
- Set every two digits for locations and to identify the data base
- Use unique method (solution of problem identified) for find the required data base.
- When searching , use specific searching algorithm to make quick and search more easier
- Here apply the parallel computation algorithm for process faster.
- This can be run parallel in the entire existing database to retrieve the massive information.

A. Advantage

Less accessing time:

All the data set are created in same manner and also those details are presented in the form of id. We can retrieve that information by id easily and also quickly. Wherever data set present we have used our id for refer that.

"our system will retrieve information parallel from data sets and display the result quickly"

B. Overview Of Proposed Method

1) Key –Matching algorithm

We have unique id which is being call as a key. Our key match with key which is presented in each data set. Once match found corresponding data will be displayed else search until match found, for that we use searching algorithm.

2) Searching Algorithm

Here we use hierarchical model for searching keys. All data set coming under one node, once we follow that node we can find our data where tat present can find easily as soon as possible.

3) Parallel Processing Algorithm

Once we entered our id it should search all data set parallel and show result in main page. It is very hard to implement. To solve this problem we need high computation power, from our point of view grid computing resources solve this problem, because it has well computation power, so it can search this all data set parallel and give result to main system.

C. Efficient Process / Work

1) Key – Matching Algorithm

We have a unique id is called as a key. That is matches with key, which is presented in each data set.

Id=a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

Once match found, the corresponding data will be display, else search until match found, so we use searching algorithm for that.

```
Int Key id;
If (key id==key id in db) then Match found;
```

Else Search goes until match found;
End;

When we enter this id in our application it should match with corresponding ID which is present in particular data set, because ID is present in data set and also in the same form. So searching is easy when we use binary search. Because, in data set all the data present in hierarchical form, once the top element is match then move into hierarchical order.

Here first match a1 with a1 for state, once it match move into their state's district only, not other's state's district. Similarly once district match move into corresponding taluk only and so on.

Once our ID till a9 is match it seems we found the address of particular holder. Once we find the address further thing will be easier only.

2) Searching Algorithm

Here we use binary and parallel search algorithm for searching keys. All data coming under one node, once we follow that node we can find our data easily as soon as possible.

Id=a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

We can search data by using keys, because we have formed data set with keys only. So searching is easy as much as possible.

If (a1==a1) Go to specific state;
If (a2==a2) Go to specific district;
....
If (a6==a6) Go to specific door no;
End;

Once we identify the address of the holder by key matching algorithm. ie. We have completed first 9 digits process and we have remaining one digit only, it may vary depends on data set.

If it's driving license its value is a10, a10, a11- ration card number, a12 - pancard number, a13 - passport number.

Last digit is optional; it means if we need anyone of above we have to go for their data set individually. And in passport number match with our id then corresponding holder's phone number, mobile number, mail-id also can retrieve.

3) Parallel Processing Algorithm

Once we entered our id it should search all data set parallel and show result in main page. It is very hard to implement. So solve this problem we need high computation power, from my point of view cluster system solve this problem, because it has well computation power, so it can search this all data set parallel and give result to main system. [2], [12], [6]

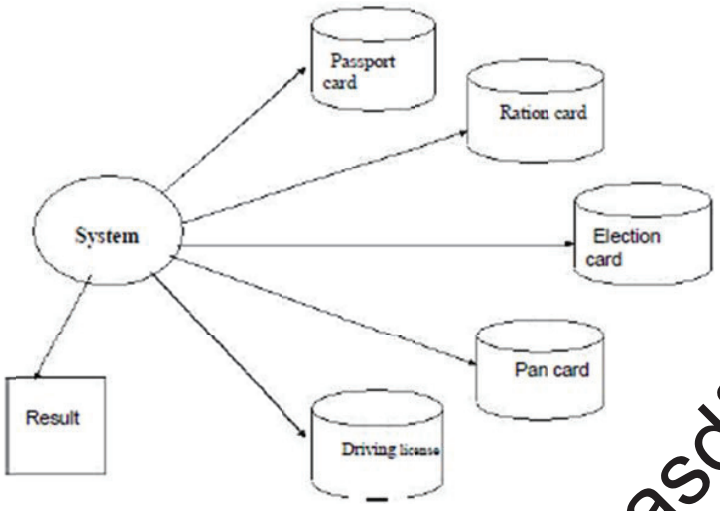

Fig.3 Parallel database

Int key id;

If (a10==a10 in license dataset) Retrieve specific no;

If (a11==a11 in ration dataset) Retrieve specific no;

If (a12==a12 in pan card dataset) Retrieve specific no;

If (a13==a13 in passport dataset) Retrieve specific no;

Display all values in system;

When we enter our id, first 9 digits show holder's information includes name, gender, address and so on. Further digits indicate driving license, ration card number, pan card number, passport number.If holder does not have passport then it should display the result. Similarly if user does not have pan card then should display that also. Once we enter our id it should match with all data set parallel and produce the result. This we need for efficiency. Our key should search the keys in all data set and match those keys and finally show the result at a time.

4) Security method

User authentication in grid computing resources is one in all the elemental procedures to confirm secure communications And share system resources over an insecure public network channel. Especially, the aim of the one-time password is to create it tougher to achieve unauthorized access to restricted resources. rather than using the password file as conventional authentication systems. various one-time password schemes using smart cards, time-synchronized token or short message service in order to cut back the risk of change of state and maintenance price. However, QR code or digital schemes are impractical because of the far from ubiquitous hardware devices or the infrastructure requirements. To remedy these weaknesses, the attraction of the QR code technique is introduced into our one-time watchword authentication protocol. This methodology can defend the resource and knowledge sets from the user and third party users. when and each level it\'ll verify the user key and validate for more security of systems / grid computing resources.[16],[17]

V. Working Model and Simulation

Authority Card

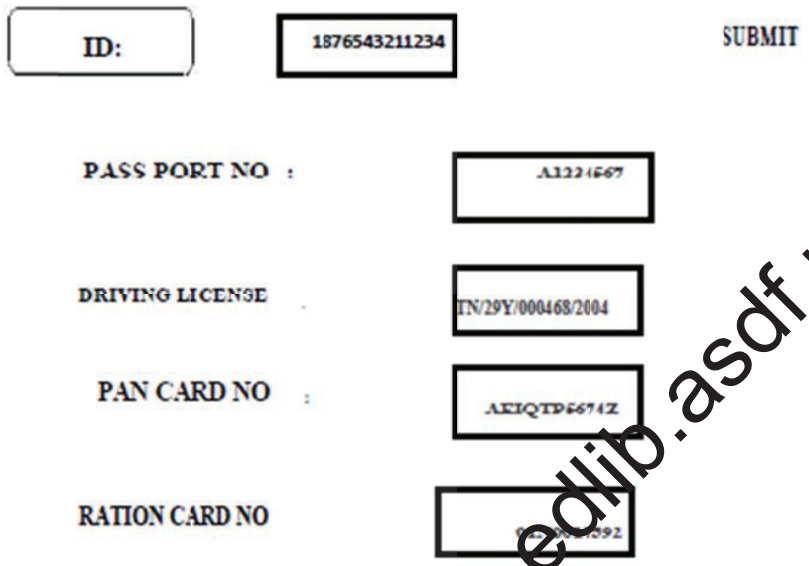

Fig. 4 Working model application

Once we enter our user ID we should click submit button. Once we submit it, the back end system will search all corresponding information at all data set parallel and produce the result.

VI. Results and Efficiency

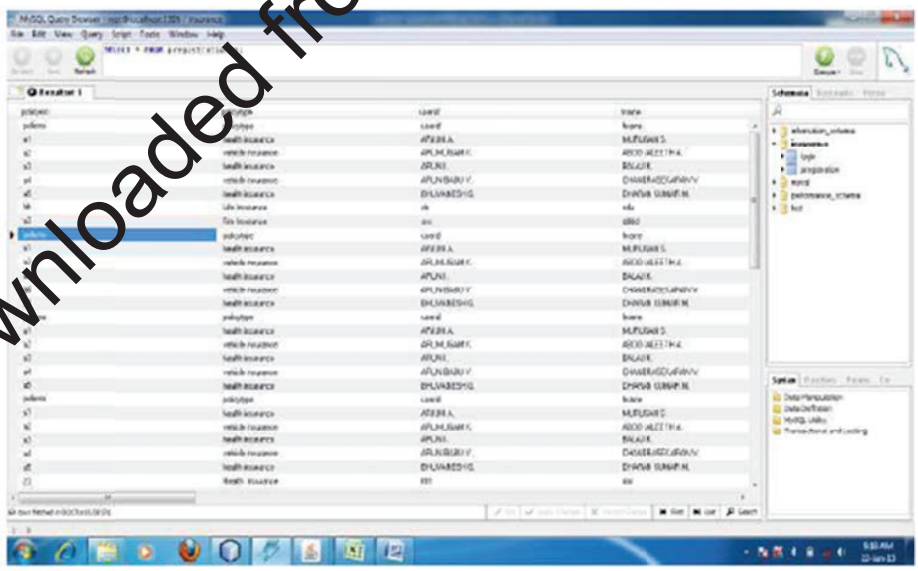

Fig.5 Data serach from multiple data sets (parallel)

Experimental results on simulated data sets (From the real data, we randomly picked 1,000 vs. 1,0000 vs. 2,000, *and* 3,000)

Algorithm	completeness	accuracy	Time(ms)	Existing solution-compared	
				Com.	Acc.
Key match	98.40%	96.10%	14593	97.70%	95.40%
search	98.40%	96.40%	13515	97.70%	95.40%
Parallel process	98.40%	96.90%	11422	97.70%	95.40%

Table 1. Experimental results on simulated data sets

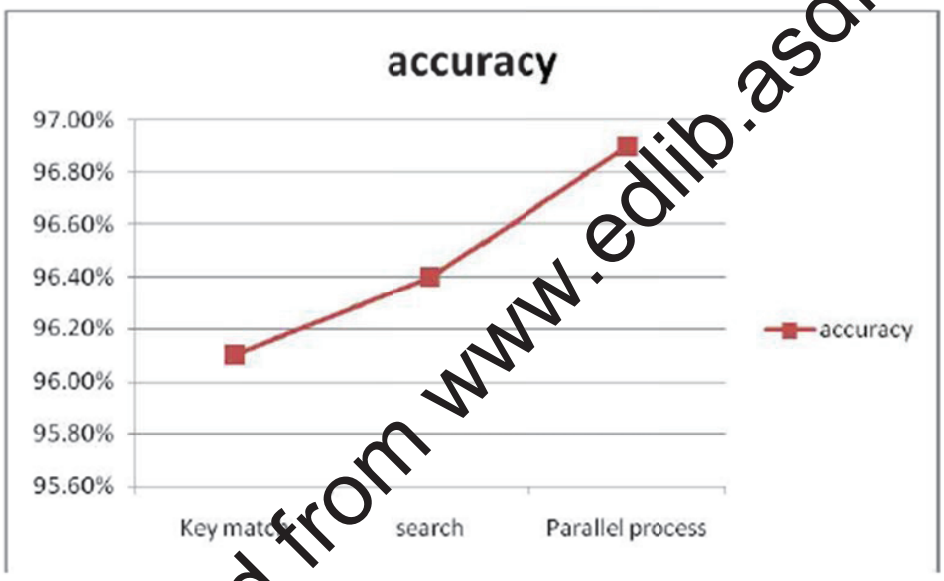

Fig.8 Efficiency (in terms of accuracy tested with sample data)

We have experimentally validated our algorithms on large simulated as well as real data. From the real data, we randomly picked 1,000 vs. 1,0000 vs. 2,000, *and* 3,000 vs. 3,000 data sets as three groups of linkage tests. Accuracy and completeness are defined stringently to show the performance of our methods.

VII. Conclusion and Future Work

With this proposed method we have executed the large data set from multiple resources simultaniously. That procedure produced result with in estimated time and less. Here integrating multiple data sets of database are efficient and that multiple data sets are compute by the grid infrastructures environment parallel.

We plan to design and develop the framework of this method for grid computing resources as a tool with security features(QR code). This framework will embedded with grid computing infrastructure as software API to increase heterogeneous data manipulations of grid data bases and it will do fast and parallel computation efficiently in all grid resources

References

1) Distributed data management for grid computing by Michael Di Stefano, 2005 by John Wiley & Sons, Inc. ISBN 0-471-68719-7.
2) Data Management and Heterogeneous Data Integration in Grid Computing Environments - K.Ashok kumar, C.Chandra sekar, Proceedings of the 2010 INCOCCI- IEEE conference..
3) Christen P. In: Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining: 24–27 August 2008; Las Vegas. Ying L, Bing L, Sunita S, editor. ACM, New York; 2008. pp. 1065–1068.
4) Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. Nature Protoc. 2009;4(1):44–57.
5) Efficient algorithms for fast integration on large data sets from multiple sources – Han Mi Sanguthevar Rajasekaran, and Robert Aseltine – BMC Medical Informatics and Decision Making Vols. 1 2013
6) Ian Foster, Jens Vo¨ckler, Michael Wilde, and Yong Zhao, "The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration," Proceedings of the 2003 CIDR Conference.
7) http://lbsitbytes2010.wordpress.com/2013/03/20/cluster-computing-2/
8) http://www.wolfram.com/gridmathematica/ integrated extension system for increasing the power of your Mathematica
9) http://www.adarshpatil.com/newsite/research.htm
10) Data integration through database federation, by L. M. Haas, E. T. Lin, M. A. Roth, IBM SYSTEMS JOURNAL, VOL 41, NO 4, 2002
11) [L].Oracle's Solution for Heterogeneous Data Integration Sponsored by: Oracle Corporation Steve McClure, August 2003.
12) K.Ashok kumar, Dr.C.Chandra sekar "Virtualization for Grid Computing Environments to increase the computing power efficiency and infrastructure", International Journal of Research in Computer Science ISSN 2249-8257 Volume 2 Issue 5 (2012) pp. 39-44
13) http://uidai.gov.in/aadhaar-technology.html
14) http://www.techrepublic.com/blog/it-security/sqrl-a-new-method-of-authentication-with-qr-codes/
15) Grid Computing Security: A Taxonomy- Chakrabarti, A. ; Infosys Technol., Bangalore ; Damodaran, A. ; Sengupta, S., Security & Privacy, IEEE (Volume:6 , Issue: 1)
16) Trusted Grid Computing with Security Binding and Trust Integration Shanshan Song, Kai Hwang, Yu Kwong Kwok , Journal of Grid Computing June 2005, Volume 3, Issue 1-2, pp 53-73
17) A One-Time Password Scheme with QR-Code Based on Mobile Phone, Kuan-Chieh Liao ; Dept. of Accounting & Inf. Syst., ASIA Univ., Taichung, Taiwan ; Wei-Hsun Lee ; Min-Hsuan Sung ; Ting-Ching Lin, *NC, IMS and IDC (NCM), 2009 Fifth International Joint Conference*