the ranking. For example, given that Destination IP addresses of certain asset systems are well known by those within an organization, this ranking would be higher

Penalty = (Delta * Ranking)/100

Finally, the chromosome's fitness is then computed using the above penalty. The scope of the fitness result is between 0 and 1.

Fitness = 1 - Penalty

Selection: Once the initial population of chromosomes is evaluated, the GA experiments with new generations and iteratively refine the initial outcomes so that those that are most fit are more probable to be ranked higher as results. The objective is to produce new generation of chromosomes to evaluate. The process of selection invdves an individual so that its offspring is copied to the next generation. The simplest way to do so, is to choose an individual based on proportional fitness (also known as roulette wheel selection). Given a weighted roulette wheel representing the percentage of individuals fitness overall population, spin the wheel and select the individual which the wheel selected.

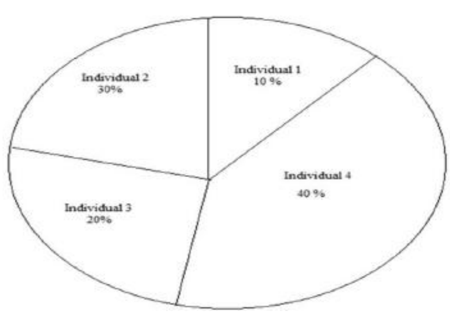

Figure 2. Reproduction based on roulette wheel selection.

The process of roulette wheel can be also described as follows: Calculate for each individual its proportion of the total population fitness, such that the sum of all the individual fitness equals 1.0. On a line between 0.0 and 1.0, the better individuals will take up a larger proportion of the line than will less fit individuals. An individual is then chosen by generating a random number between 0.0 and 1.0 and then selecting the individual in whose region of the number line the random number falls. By repeating this each time-an individual needs to be chosen, the better individuals will be chosen more often than the poorer ones, thus fulfilling the requirements of survival of the fittest.

Crossover: In essence, the crossover operation creates new chromosomes that share optimistic characteristics of the parent chromosomes while at the same time lowering the negative attributes in a child chromosome . Figure 3 provides an example of a crossover of chromosomes from the parents to their offspring. Although this step is typical in most genetic algorithms, in the case of this project's chromosome

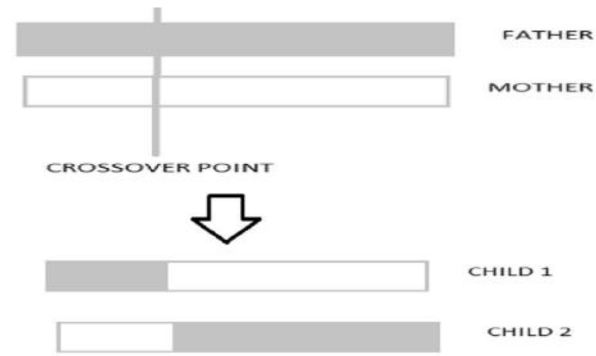

Figure 3. Crossover Operator.

(Table 4) the crossover operation may not be beneficial as crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. In reproduction, good strings in a population are probabilistic-ally assigned a larger number of copies and a mating pool is formed.

It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool. The two strings participating in the crossover operation are known as parent strings and the resulting strings are known as children strings, if an appropriate site is chosen. With a random site, the children strings produced may or may not have a combination of good sub-strings from parent strings, depending on whether or not the crossing site falls in the appropriate place. But this is not a matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover.

It is clear from this discussion that the effect of cross over may be detrimental or biennial. Thus, in order to preserve some of the good strings that are already present in the mating pool, all strings in the mating pool are not used in crossover. A crossover operator is mainly responsible for the search of new strings even though mutation operator is also used for this purpose sparingly. Many crossover operators exist in the GA literature. One site crossover and two site crossover are the most common ones adopted. In most crossover operators, two strings are picked from the mating pool at random and some portion of the strings is exchanged between the strings. Crossover operation is done at string level by randomly selecting two strings for crossover operations. A one site crossover operator is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site.

While a Source or Destination IP may be bound by upper and lower IP settings, a crossover of the IP octet values would probabilistically not be advantageous. For example, the crossover of the parental values of 209.103.51.134 and 101.1.25.193 could result in child IP addresses of 209.103.25.193 and 101.1.51.134. However, the probability that this offspring will be potential suspicious Source or Destination IP addresses is low.

Mutation: Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes of individuals to be different from those of their parent individuals. Mutation in a way is the process of randomly disturbing genetic information.

They operate at the bit level; when the bits are being copied from the current string to the new string, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability p_m. The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population

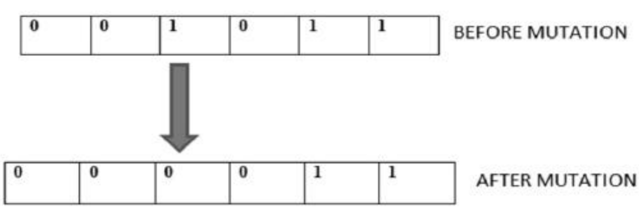

Figure 4. Mutation Operator.

VIII. RESULT AND ANALYSIS

From the system we get the confusion matrics depicted in table 4. For most of the classes, this system performs well enough except normal data type which is because of ignoring on numerical features. We get better detection rate for denial of service & user-to-root and close detection rate for probe & remote-to-local.

TABLE 4
CONFUSION METRICS FOR SYSTEM EVALUATION

		Predicted label					% correct
		Normal	Probe	Dos	U2r	R2l	
Actual class	Normal	421	142	1583	486	713	69.5
	Probe	398	296	654	2	129	71.1
	Dos	521	432	228	1	10	99.4
	U2r	146	21	8	43	10	18.9
	R2l	119	57	339	141	881	5.4
% correct		76.9	54.7	92.0	64	50	

For simplified evaluation of our system, besides the classical accuracy measure, we have used two standard metrics of detection rate and false positive rate developed for network intrusions. Table 5 shows these standard metrics.

TABLE 5
STANDARD METRICS FOR SYSTEM EVALUATION

		Predicted label	
		Normal	Intrusion
Actual Class	Normal	True Negative (42138)	False Positive (18455)
	Intrusion	False Negative (12528)	True Positive (237908)

Detection rate for each data type can be seen from figure 5.

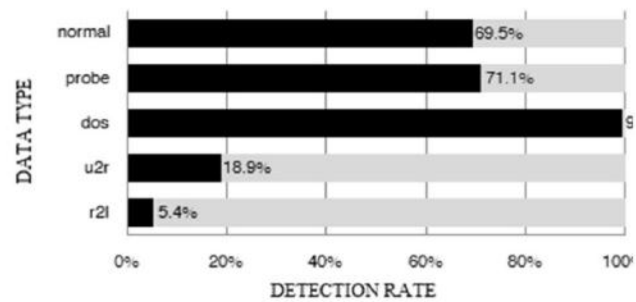

Figure 5. Detection rate for each class.

Detection Rate: Detection rate (DR) is calculated as the ratio between the number of correctly detected intrusions and the total number of intrusions [1], that is

$$DR = \frac{\#TruePositive}{\#FalseNegative + \#TruePositive}$$

for set of 100 attempts DR = 0.9500.

False Positive Rate: False positive rate (FP) is calculated as the ratio between the numbers of normal connections that are incorrectly classified as intrusions and the total number of normal connections [1], that is

$$\mathrm{FP} = \frac{\#FalsePositive}{\#TrueNegative + FalsePositive}$$

FP = 0.3046.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, authors presented and implemented an Intrusion Detection System by applying genetic algorithm to efficiently detect various types of network intrusions. Genetic algorithms usually use the population diversity as the performance measure for a number of reasons such as avoiding premature convergence. Authors proposed techniques for measurement of population diversity based on the phenotype and genotype properties. The main contribution of this study is the crossover operator and probability selection technique based on the population diversity using fuzzy logic controllers.

In near future, this intrusion detection system can be improved with the help of more statistical analysis and with better and may be more complex equations. In addition to this, different type of genetic algorithm such as ZCS, MMSGBML etc. can be applied to design more efficient system.

X. REFERENCES

[1]. Alberto Fern´ andez, Salvador Garcıa, Julian Luengo, Ester Bernad´ **o-**Mansilla, " *IEEE Trans. Evolutionary Computation"* Volume 14, No. 6, December 2010.

[2]. Marco Dorigo and UweSchnepf *IEEE Trans. Systems, Man, and Cybernetics",* Volume 23, January 1993.

[3]. Tommkovas, "*Genetic Based Machine Learning*", University Of Bristol Handbook.

[4]. Wafa' S.Al-Sharafat and ReyadhNaoum" Adaptive Framework for Network Intrusion Detection by Using Genetic-Based Machine Learning Algorithm " world academy of science, engineering and Technology.

[5]. F. Djannaty and S. Doostdar, "A hybrid genetic algorithm for the Multidimensional knapsack problem," International Journal of Contemporary Mathematical Sciences, volume 3, no. 9, pp. 443–456, 2009.

[6]. G. Sywerda, "Uniform Crossover in Genetic Algorithms," in *Proc 3rd International Conference on Genetic Algorithms*, pp. 2–9, Morgan Kaufmann, 1989.

[7]. J. P. Planquart, *Application of Neural Networks to Intrusion Detection*, SANS Institute.

[8]. R. G. Bace, *Intrusion Detection*, Macmillan Technical Publishing. 2000.

[9]. V. Bobor, *Efficient Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithm*, Department of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, KTH/DSV, 2006.

[10]. KDD-CUP-99 Task Description; http://kdd.ics.uci.edu/databases/kddcup99/task.html

[11]. KDD Cup 1999: Tasks; http:// www. kdd.org/ kddcup/index.php? section=1999&method=task

[12]. H. G. Kayacık, A. N. Zincir-Heywood, M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", May 2005.

[13]. G. Folino, C. Pizzuti, G. Spezzano, "GP Ensemble for Distributed Intrusion Detection Systems". ICAPR 54-62, 2005.

[14]. Sectools.Org: 2006 Results; http://sectools.org/tools2006.html

[15]. SecTools.Org: Top 125 Network Security Tools; http://sectools.org/tag/ids/

[16]. Snort (software); http://en.wikipedia.org/wiki/Snort_%28software %29

[17]. InfoWorld, The greatest open source software of all time, 2009;

[18]. The Bro Network Security Monitor; http://bro-ids.org/

[19]. K. Scarfone, P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)". Computer Security Resource Center (National Institute of Standards and Technology). February 2007.

[20]. A. Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms", January 2005.

Binu Kaushik received M. Tech degree from Maharishi Dayanand University in computer science and engineering and currently working as an assistant professor of Information Technology in IEC Group of institutions, Greater Noida since 2010. Presented many papers in national and international conference on soft computing. She has abiding interest for teaching and research.

Shashi Bhushan Vishwakarma obtained B.Tech (CSE) degree in 2008 and M.Tech (IT) degree in 2012. Currently he is teaching in IEC Group of institutions, Greater Noida as an Assistant Professor of Information Technology. Presented papers in national and international conferences on soft computing and other topics.

Data Management and Heterogeneous Data Integration in Grid Computing Environments

Rishabh Rai, MIEEE
Department of Electronics and Communication Engineering, Vidyadaan Institute of Technology and Management, Ariaon, Buxar 802119 Bihar India
rishabh.rahul001@gmail.com

Abstract: **Ensembles of distributed, heterogeneous resources, or Computational Grids, have emerged as popular platforms for deploying large-scale and resource-intensive applications. Large collaborative efforts are currently underway to provide the necessary software infrastructure. Grid computing raises challenging issues in many areas of computer science, bioinformatics, high energy physics and especially in the area of distributed computing, as Computational Grids cover increasingly large data networks and span many organizations. We choose to focus on issues concerning the dissemination and retrieval of information from distributed networks and data integration on Computational Grid platforms. This paper is of interest to distributing computing researchers because Grid computing provides new challenges that need to be addressed, as well as actual platforms for experimentation and research.**

Keywords: Grid computing, Data management, Data integration, Heterogeneous resources.

I. INTRODUCTION

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.

Grid computing harnesses a diverse array of machines and other resources to rapidly process and solve problems beyond an organization's available capacity. Academic and government researchers have used it for several years to solve large-scale problems, and the private sector is increasingly adopting the technology to create innovative products and services, reduce time to market, and enhance business processes.

The term grid, however, may mean different things to different people. To some users, a grid is any network of machines, including personal or desktop computers within an organization. To others, grids are networks that include computer clusters, clusters of clusters, or special data sources. Both of these definitions reflect a desire to take advantage of vastly powerful but inexpensive networked resources. In our work, we focus on the use of grids to perform computations as opposed to accessing data, another important area known as data grid research [8].

B. Different systems

Grid computing is akin to established technologies such as computer clusters and peer-to-peer computing in some ways and unlike them in others. Peer-to-peer computing, for example, allows the sharing of files, as do grids, but grids enable users to share other resources as well. Computer clusters and distributed computing require a close proximity and operating homogeneity; grids allow computation over wide geographic areas using computers that are heterogeneous.

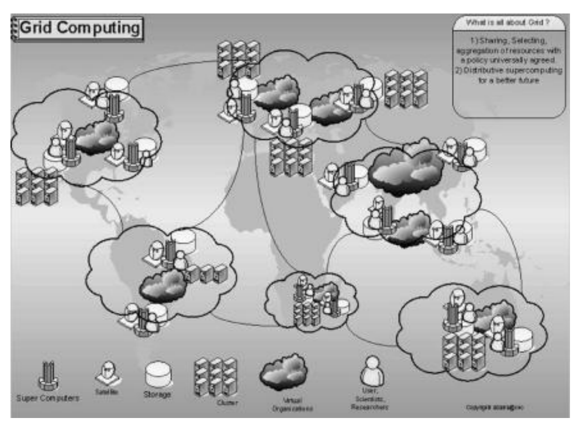

Figure 1. Grid computing.

C. Grids are usually heterogeneous Networks & Data

Grids are usually heterogeneous networks. Grid nodes, generally individual computers, consist of different hardware and use a variety of operating systems, and the networks connecting them vary in bandwidth. Realizing the vision of ubiquitous parallel computing on a grid will require that we make grids easy to use, and this need applies both to the creation of new applications and to the distribution and management of applications on the grid itself. To accomplish this goal, we need to establish standards and protocols such as open grid services architecture—which allows communication across a network of heterogeneous machines—and tool kits such as Globus, which implement the rules of the grid architecture.

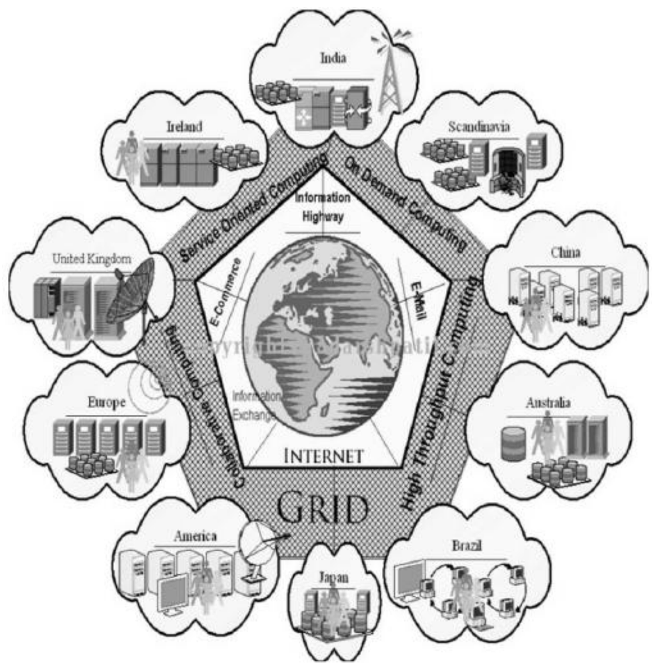

Figure 2. Heterogeneous Networks.

II. DATA MANAGEMENT

Data Services
A grid fundamentally consists of two distinct parts, compute and data:

A. Compute grid—provides the core resource and task management services for grid computing: sharing, management, and distribution of tasks based on configurable service-level policies.

B. Data grid—provides the data management features to enable data access, synchronization, and distribution of a grid.

Efficient access to and movement of huge quantities of data is required in more and more fields of science and technology. In addition, data sharing is important, for example enabling access to information stored in databases that are managed and administered independently. In business areas, archiving of data and data management are essential requirements.

C. Objectives
Data services are used to move data to where it is needed, manage replicated copies, run queries and updates, and transform data into new formats. They also provide the capabilities necessary to manage the metadata that describes OGSA data services or other data, in particular the provenance of the data itself.

1) Data services requirements include:
- *Data access.* Easy and efficient access to various types of data (such as database, files, and streams), independent of its physical location or platform, by abstracting underlying data sources is required. Mechanisms are also required for controlling access rights at different levels of granularity.

- *Data consistency.* OGSA must ensure that consistency can be maintained when cached or replicated data is modified.

- *Data persistency.* Data and its association with its metadata should be maintained for their entire lifetime. It should be possible to use multiple persistency models.

- *Data integration.* OGSA should provide mechanisms for integrating heterogeneous, federated and distributed data. It is also required to be able to search data available in various formats in a uniform way.

- *Data location management.* The required data should be made available at the requested location. OGSA should allow for selection in various ways, such as transfer, copying, and caching, according to the nature of data[4].

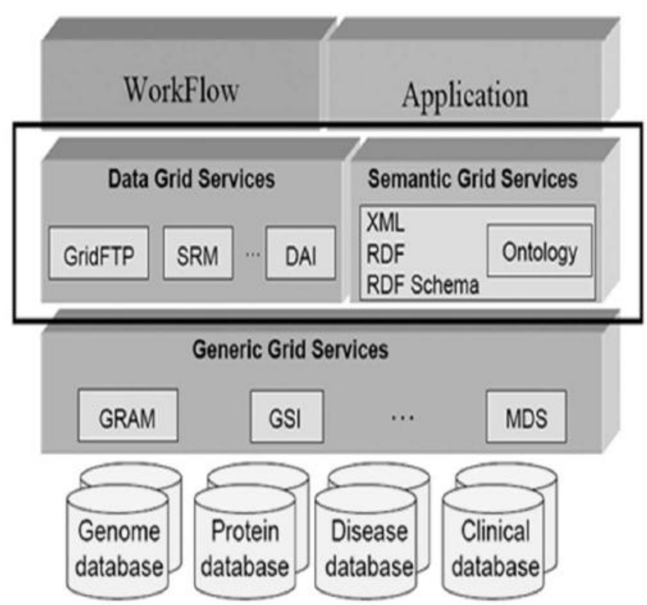

Figure 3. Data Management.

III. FTP IN GRID (GRID FTP)

The File Transfer Protocol (FTP) is one of the most widely used protocols for the movement of files across a network. It is amazing that it remains in heavy use even in today's technology advanced society. Therefore, it is an obvious choice for data movement within a grid environment. The standards body

Globus is investigating the use of FTP as the data transfer protocol for a data grid implementation, termed GridFTP.

Grid FTP is a protocol defined by Global Grid Forum Recommendation GFD.020, RFC 959, RFC 2228, RFC 2389, and a draft before the IETF FTP working group. The Grid FTP protocol provides for the secure, robust, fast and efficient transfer of (especially bulk) data. The Globus Toolkit provides the most commonly used implementation of that protocol, though others do exist (primarily tied to proprietary internal systems). [2], [3] [5].

IV. DIFFERENT IMPLEMENTATIONS OF A GRID

A. Level 0 Data Grids

Level 0 data grids were the earliest to address data requirements in a grid topology.

Their main function is the distribution of large, static data sets to the nodes in the grid. They do not address data management issues such as updates, transactions, or integration with external systems, as illustrated by the following academic examples. The first example is found in the white paper by Chervenak *et al*. as quoted below [1].

In an increasing number of scientific disciplines, large data collections are emerging as important community resources. In this paper, we introduce design principles for a data management architecture called the Data Grid. We describe two basic services that we believe are fundamental to the design of a data grid, namely, storage systems and metadata management. Next, we explain how these services can be used to develop higher-level services for replica management and replica selection. We conclude by describing our initial implementation of data grid functionality.

Another similar argument is presented in the white paper by Moore *et al.* [7].

Data grids link distributed, heterogeneous storage resources into a coherent data management system. From a user perspective, the data grid provides a uniform storage of name space across the underlying storage systems, while supporting retrieval and storage of files. In the high energy physics community, at least six data grids have been implemented for the storage and distribution of experimental data. Data grids are also being used to support projects as diverse as digital libraries (National Library of Medicine: Visible Embryo project), federation of multiple astronomy sky surveys (NSF National Virtual Observatory project), and integration of distributed data sets (Long Term Ecological Reserve). Data grids also form the core interoperability mechanisms for creating persistent archives, in which data collections are migrated to new technologies over time. The ability to provide a uniform name space across multiple administration domains is becoming a critical component of national-scale, collaborative projects.

B. Level 1 Data Grids

Level 1 data grids support data sets that are dynamic in nature: data sets that change daily, hourly, minute-to-minute, second-to-second, or at any other intervals. Level 1 data grids address the distribution of and the ready access to data across the many nodes of the compute grid. They supply, among other things

- Access methods
- Management methods
- Transactional methods
- Synchronization methods.

V. DATA INTEGRATION

Data integration takes many forms, from simple file transfers to virtual database platforms. According to IDC [11], data integration software attempts to provide noninvasive[16], programmatic access to persistent structured data, whether in heterogeneous, homogenous, distributed, or centralized data sources. Organizations implementing data integration solutions would do so for one or more of the following reasons:

A. Provide an integrated view of data. Such a view may be in support of data usage beyond tha governing the creation and maintenance of the data in its source. Data warehousing and business analytics fit here, as do enterprise information portals. The primary business benefit is a unified view of organizational information for business analytics and decision making [13].

*B. Allow multiple applications to behave cooperatively and harmoniously***.** This is typically done using a messaging strategy where a consolidated resource queue publishes common messages, to which consumers of the queue can subscribe. The primary business benefit is a higher level of customer service and improved operational efficiency [10].

*C. Improve operational efficiency of the IT department***.** Efficiency is achieved by consolidating the number of different data sources maintained, creating a unified virtual (or federated) database for use with new applications, or implementing a unified system for information sharing. The primary business benefit is improved developer efficiency, resulting in less investment in equipment and staff and more rapid response to changes in the business environment.

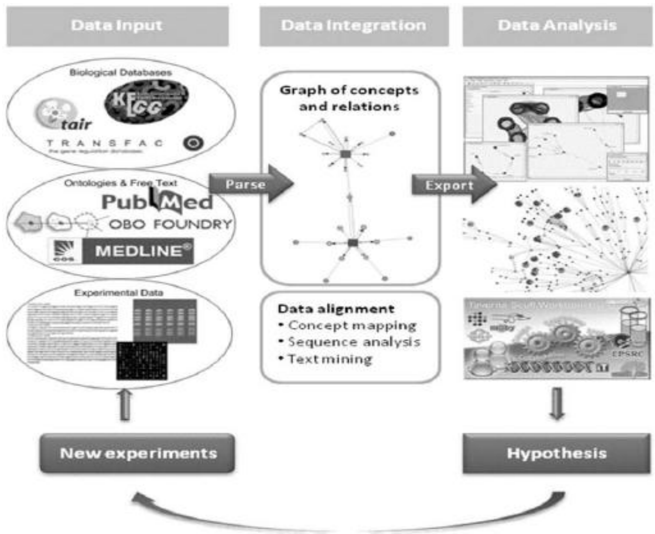

Figure 4. Data Integration.

VI. INTEGRATION CHALLENGES

Although, as stated above, a majority of IT managers rated integration as either extremely important or critical, adoption has been slow. In a survey completed in 2001, IDC asked technology and business professionals which of the following strategies they used to integrate ecommerce or call center applications with back-office or front-office systems:

- Standalone (*i.e.*, no integration)
- File transfers
- File transfer with queued data
- Bidirectional replication
- Messaging
- Transaction messaging
- Transaction messaging with data synchronization.

VII. INTEGRATION STRATEGIES

Clearly, given this range of requirements, there are a variety of different integration Strategies, including the following:

A. Consolidated. A consolidated data integration solution moves all data into a single database and manages it in a central location.

B. Federated. A federated data integration solution leaves data in the individual data source where it is normally maintained and updated and simply consolidates it on the fly as needed. In this case, multiple data sources will appear to be integrated into a single virtual database, masking the number and different kinds of databases behind the consolidated view. These solutions can work directionally.

C. Shared. A shared data integration solution actually moves data and events from one or more source databases to a consolidated resource, or queue, created to serve one or more new applications. Data can be maintained and exchanged using technologies such as replication, message queuing, transportable table spaces, and FTP.

VIII. DATA INTEGRATION SOLUTION

The Oracle9*i* RDBMS is at the center of Oracle's support for data integration. Oracle9*i* includes the features, functions, and capabilities that enable an organization to integrate its data regardless of where or how it is maintained.

With the exception of Oracle Transparent Gateway, the Oracle data integration features are integrated with Oracle9*i*, allowing an organization to efficiently adapt the Oracle9*i* capabilities to fit its specific needs. The result is the attainment of data integration benefits, such as faster time to market, with less development effort and lower total cost of ownership (TCO).

1. Implementing federated data integration
2. Implementing data integration for data sharing
3. Dealing with heterogeneous data sources
4. Enabling integrated search of data and content with Ultra Search.

At the same time, other database management products such as DB2, Sybase, the SAP file system, flat files, Web services, or other data types may also be included in the mix of data sources to be integrated. In every case, the data integration solution will rely on functions and features of the Oracle9*i* RDBMS.

It should also be noted that in those cases where packaged enterprise applications are being integrated, a comprehensive enterprise application integration (EAI) platform, which uses features such as those found in Oracle9*i*AS, will be employed.

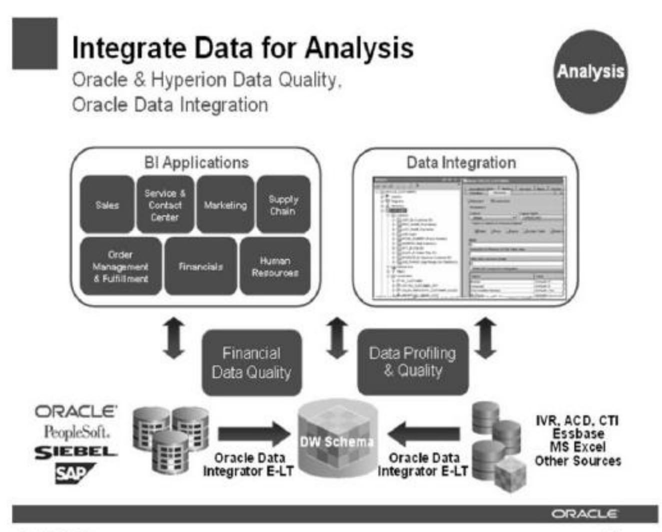

Figure 5. Oracle – Data Integration & Analysis.

IX. DEALING WITH HETEROGENEOUS DATA SOURCES

A heterogeneous environment for the purposes of this discussion is one involving one or more of the following: non-Oracle data sources, non-Oracle message-queuing software, or non-SQL applications.

In other words, environments where Oracle software must interoperate with other vendors' software. To gain the promised benefits of Oracle's data integration solutions, this interoperability needs to be achieved as transparently as possible so that application developers don't have to customize their applications to deal with heterogeneous data sources (*i.e.,* they can build on one consistent interface) [14].

A. Non-Oracle data sources: Oracle provides Transparent Gateways to achieve transparent interoperability with other major RDBMSs, such as DB2, SQL Server, and Sybase. In addition, it offers a Generic Connectivity feature for interoperability via ODBC and OLE DB. This allows for access to data stores for which Oracle does not have a Transparent Gateway.

B. Non-Oracle message-queuing software: Oracle provides the Messaging Gateway feature to support communication between Oracle Streams and other non-Oracle message-queuing systems, such as IBM's MQ Series.

C. Non-SQL applications: Oracle's approach to this requirement is to offer a variety of open interfaces with which users can interoperate with third-party applications or allow users to access Oracle9*i* databases from their own client applications.

X. CONCLUSION

We have discussed motivation of data integration for heterogeneous data. This heterogeneous data integration or data integration involved in large data set of bioinformatics sequence analysis and molecular. During this analysis we got large amount of data (heterogeneous) and these are integrated by oracle 9i, 11g and enterprise manager. This approach can use for various large set of data integration and its information access, like country's national identification card, statistical and high performance, etc.

XI. REFERENCES

[1]. Protecting File Systems: A Survey of Backup Techniques, Ann L. Chervenak , Vivekanand Vellanki and Zachary Kurmas, fannc, vivek, kurmaszg@cc.gatech.edu

[2]. http://globus.org/toolkit/

[3]. http://dev.globus.org/wiki/Welcome#Data_Management

[4]. http://www.globus.org/toolkit/docs/4.0/data/key/index.html#id2501317.

[5]. http://dev.globus.org/wiki/GridFTP.

[6]. Globus Grid FTP: What's New in 2007 (Invited Paper). John Bresnahan, Michael Link, Gaurav Khanna, Zulfikar Imani,

Figure 6. Oracle – Data Integration (result window).

Rajkumar Kettimuthu and Ian Foster. *Proc.First International Conference on Networks for Grid Applications* Oct, 2007.
[7]. Distributed data management for grid computing by Michael Di Stefano, 2005, John Wiley & Sons, Inc.
[8]. The Grid 2: Blueprint for a New Computing Infrastructure , Edited by Ian Foster and Carl Kesselman , Morgan Kaufmann; 2 edition, November 18, 2003.
[9]. Ian Foster, Jens Vo¨ckler, Michael Wilde and Yong Zhao, "The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration," *Proc 2003 CIDR Conference,* p. 1 (http://www.griphyn.org/chimera/papers/CIDR.VDG.crc.submitted.pdf).
[10]. Artemis: Integrating Scientific Data on the Grid. R. Tuchinda, S. Thakkar, Y. Gil, E. Deelman. *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence*, July 2004.
[11]. http://www.oracle.com/technology/products/ultrasearch/html/ultrasearch_architecture.htm
[12]. http://otn.oracle.com/products/ultrasearch/content.html.
[13]. L. M. Haas, E. T. Lin, M. A. Roth*, IBM Systems Journal,* Data integration through database federation, Volume 41, No 4, 2002
[14]. Oracle's Solution for Heterogeneous Data Integration, Sponsored by: Oracle Corporation Steve McClure, August 2003.
[15]. http://www.oracle.com/technology/software/products/database/oracle11g/112010_linx8664soft.html.
[16]. http://enterprise-manager.blogspot.com/2010/05/oracle-enterprise-manager-11g-grid.html.,
[17]. http://blogs.oracle.com/dataintegration/

Rishabh Rai is currently working as an Assistant Professor since July, 01, 2016 in Department of Electronics and Communication Engineering at Vidyadaan Institute of Technology and Management, Ariaon, Buxar, Bihar (affiliated to Aryabhatta Knowledge University, Patna, Bihar). His areas of interest are Digital System Design, Communication Systems, Embedded System Design and Low Power V.L.S.I. Design. He completed M.Tech (2013-15) from Ajay Kumar Garg Engineering College, Ghaziabad in VLSI Design with Honours in 2015 and B.Tech (2009-13) from Vishveshwarya Institute of Engineering and Technology, Gr. Noida (affiliated to Uttar Pradesh Technical University, Lucknow, U.P.) in Electronics and Telecommunication Engineering with Honours in 2013.

Appreciated and certified for academic performance & excellence in the respective branch for consecutive four years during B.Tech and Amul Vidya Bhushan Award for his academic performance & excellence in A.I.S.S.C.E. - 2009, at the District Level. Awarded with the Best Paper Presentation in the I.E.E.E. Sponsored National Conference, 'E.T.E.A.T. – 2013'. Was selected for the successful completion of his project "Face Detection Using MATLAB – (An Image Processing Based)"in the 6th Science Conclave at Indian Institute of Information Technology, (I.I.I.T.), Allahabad, U.P. in 2013.

Published 17 papers. He is the Editor-in-Chief of the Journal Sankalan: since July, 2015. He is a member of IEEE.