Heterogeneous Data Integration, Data Extraction and Data Management using Grid Computing

Manikandan. T, M.E [1], Kumar Utkarsh, (B.E) [2]

[1] Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan College of Engineering and Technology, Anna University, Chennai, India,

[1]stsmani@gmail.com

[2] Final Year Student, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan College of Engineering and Technology, Anna University, Chennai, India,

[2]utkarsh200@rediffmail.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. In this paper we briefly motivate Grid computing and introduce its basic concepts. We then highlight a number of distributed computing research questions, and discuss both the relevance and the short-comings of research results when applied to Grid computing. We choose to focus on issues concerning the dissemination and retrieval of information from distributed networks and data integration on Computational Grid platforms. We feel that these issues are particularly critical at this time, and as we can point to preliminary ideas, work, and results in the Grid community and the distributed computing community. 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, Data Extraction, Heterogeneous resources.**

I. INTRODUCTION

A. Grid computing

There have 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 [1].

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.

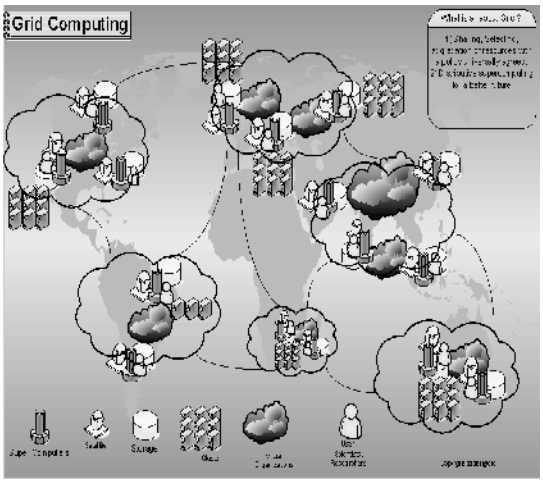

Fig.1 Grid computing

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.

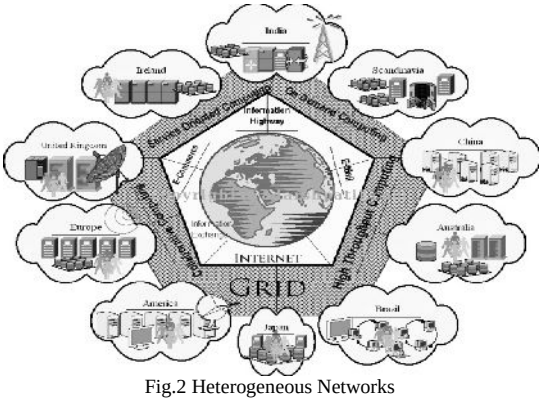

Fig.2 Heterogeneous Networks

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].

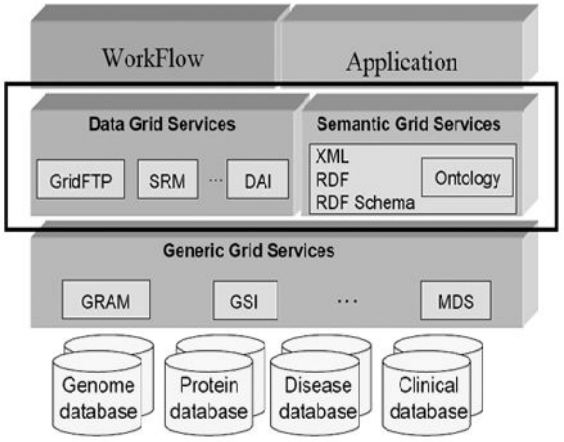

Fig.3 Data Management

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.

GridFTP 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 GridFTP 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 IMPLEMENTATION 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.8 as quoted below: [8]

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 called 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 [L] [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 that 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.

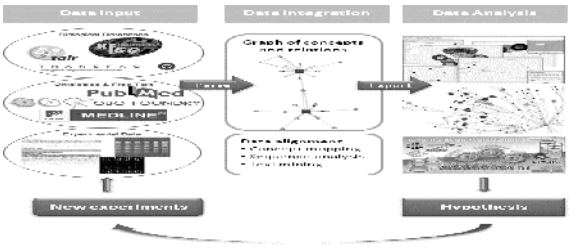

Fig.4 Data Integration

VI. DATA EXTRACTION

Data extraction is the act or process of retrieving data out of (usually unstructured or poorly structured) data sources for further data processing or data storage (data migration). The import into the intermediate extracting system is thus usually followed by transformation and possibly the addition of metadata prior to export to another stage in the data workflow.

Fig.5 Data extraction

Usually, the term data extraction is applied when (experimental) data is first imported into a computer from primary sources, like measuring or recording. Today's electronic devices will usually present an electrical connector (e.g. USB) through which 'raw data' can be streamed into a personal computer.

Typical unstructured data sources include web pages, emails, documents, PDFs, scanned text, mainframe reports, spool files etc. Extracting data from these unstructured sources has grown into a considerable technical challenge where as historically data extraction has had to deal with changes in physical hardware formats, the majority of current data extraction deals with extracting data from these unstructured data sources, and from different software formats. This growing process of data extraction from the web is referred to as Web scraping. [17].

VII. 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

VIII. 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 bidirectionally.

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.

IX. 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.

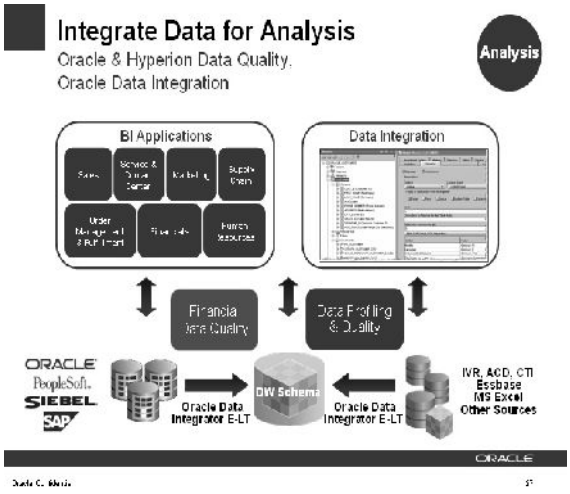

Fig.6 Oracle – Data Integration & Analysis

X. 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 Oracle9i databases from their own client applications.

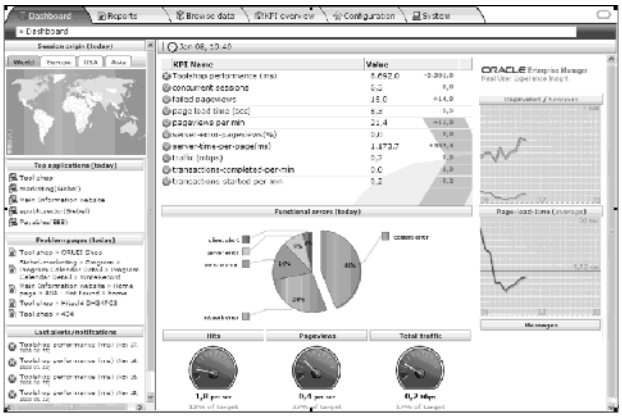

Fig.7 Oracle – Data Integration (result window)

XI. CONCLUSIONS

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.,

XII. ACKNOWLEDGEMENT

We acknowledge with our deep sense of gratitude to our honourable Chairman Thiru. A. Srinivasan, who gave us this temple of learning.

We are also grateful to our respected Principal Dr. R. Ponraj and our respected Vice-Principal Mr. Pon. Arivanantham, who regularly encouraged and guided us to complete the paper.

We are thankful to our Head of the Department of Computer Science and Engineering Mr. N. Thulasi for continuously supporting us with all the resources.

We cannot forget to thank our co-staff and friends who regularly encouraged and supported us to reach this landmark.

REFERENCES

1. The Grid 2: Blueprint for a New Computing Infrastructure , Edited by Ian Foster and Carl Kesselman , Morgan Kaufmann; 2 edition (November 18, 2003), ISBN: 1558609334.
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 GridFTP: What's New in 2007 (Invited Paper). John Bresnahan, Michael Link, Gaurav Khanna, Zulfikar Imani, Rajkumar Kettimuthu and Ian Foster. *Proceedings of the First International Conference on Networks for Grid Applications (GridNets 2007)*, Oct, 2007.
7. Distributed data management for grid computing by Michael Di Stefano, 2005 by John Wiley & Sons, Inc. ISBN 0-471-68719-7
8. Protecting File Systems: A Survey of Backup Techniques, Ann L. Chervenak , Vivekanand Vellanki , Zachary Kurmas, fannc,vivek,kurmaszg@cc.gatech.edu
9. 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, 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. Data integration through database federation, by L. M. Haas ,E. T. Lin, M. A. Roth, IBM SYSTEMS JOURNAL, VOL 41, NO 4, 2002

14. [L].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. , http://blogs.oracle.com/dataintegration/

17. http://en.wikipedia.org/wiki/Data_extraction