Grid Computing: An Introduction of high Performance Computational Environment

[1]Sandeep Kumar Verma, [2]Turendar Sahu, [3]Mohit Shakya, [4]Raksha Pandey
[1,2,3]B.Tech Student, Department of Computer Science and Engineering, Institute of Technology Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G), India
[4]Assistant Professor, Department of Computer Science and Engineering, Institute of Technology Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G), India

Abstract : Grid computing environment is used for large computation problem. In Today's world various complex tasks has been done in different scientific area so there is need of a lot of computational power to solve different types of complex scientific problems. Although Humans have good brain but still can not able to solve complex scientific problems as computers can. Grid's High performance computational environment is a good Solution. Performance of Grid is Based on various Constraints as the resource's bandwidth, resource's computational power, File size of jobs, the locations of components and so on. In this paper we are giving brief introduction of Grid Computing and some simulation tools used in Grid Area.

I. INTRODUCTION

The computer has evolved very much and became very versatile gadget and still evolving day by day. The computer has touched various fields of science and became a daily use machine. From the development of first computer (i.e. an automatic machine) a question in computer science has arrived that How can we build a powerful computer which would be capable of solving the huge and complex problem easily and quickly? In last few decades many approaches have been implemented such as distributed computing, parallel computing, sharing of hardware and software to get the answer of above mentioned question. The development of computer has given birth to many concepts such as Distributed System, Networking, etc. to empower the functionality of the computing. In [1,2], the Distributed System has many concepts and fields from which some of them are Cluster Computing, Cloud Computing, GRID Computing. Our main concern is GRID Computing. GRID computing is evolving very sharply day by day. The 'GRID' is the term, often present as an analogy to a power grid where one can get the power or electricity without concerning about its source [3][4][5]. It is also related to the Networking [6] through which one can possess great power same as supercomputer [7,8] in his/her computer i.e. power of many computers within convenient cost. In [9, 20] In 1990s many scientists were finding a way to build a powerful computer and they came with a solution known as GRID Computing. The GRID computing can be considered as a very huge collection of computers which are geographically present at various different locations all over the world. These computers mays be of different architecture [10], capacity, and of different resource set or something else. In [11], The GRID users are of two types first the user who is sharing his/her resources has a keen eye in gaining the profit by sharing his/her and the second user who is actually looking for access to the GRID resources in convenient prize. GRID computing can be considered as the journey [3] with the path of integrity of various technologies and remedies which leads toward the final goal. The key point of GRID computing is that the solution is obtained by the co-ordination of various geographically dispersed computing power or gridlets.

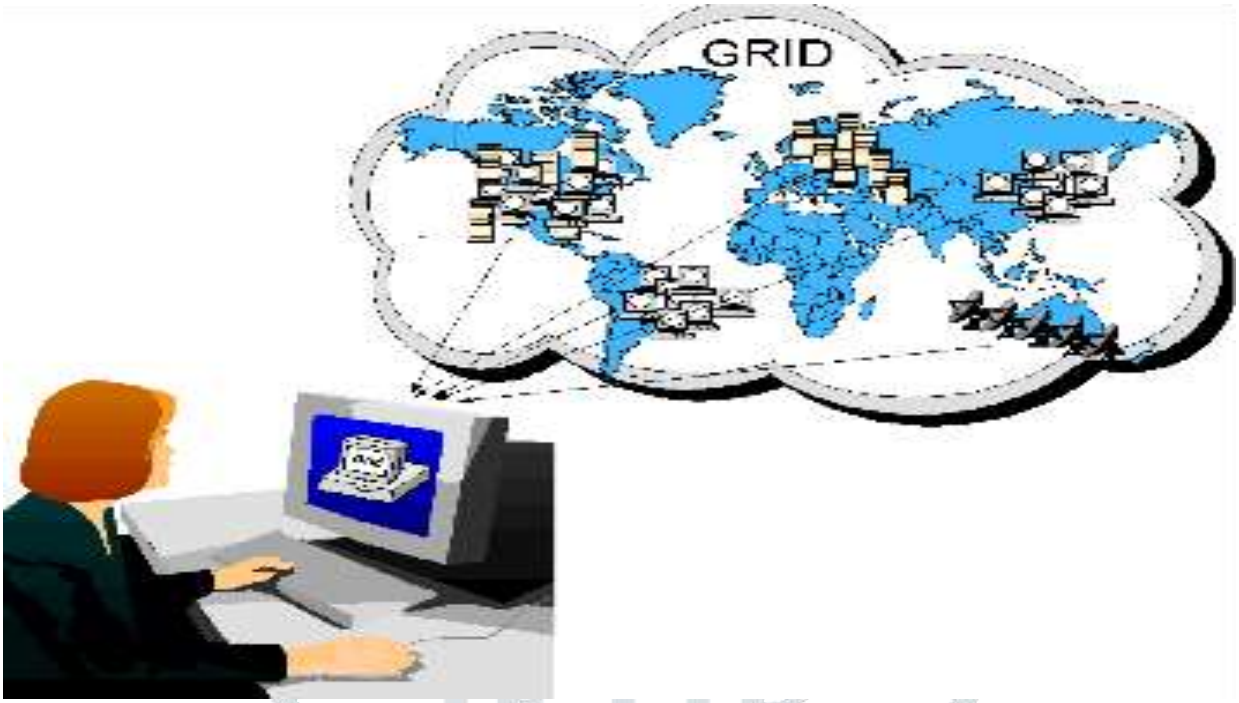

Figure 1: The grid virtualizes heterogeneous, geographically disperse resources, In [2].

II. ARCHITECTURE OF GRID ENVIRONMENT

As we have mentioned above that in GRID computing many systems which are part of environment is physically present in different locations all over the world. These system or computers form virtual organization. The user who want to process his/her job will send the job to the grid environment. These jobs are organised and send by his scheduler which is present in his/her local machine or computer. The scheduler schedules the jobs in very efficient way in order process or utilize the available resources as maximum as possible.

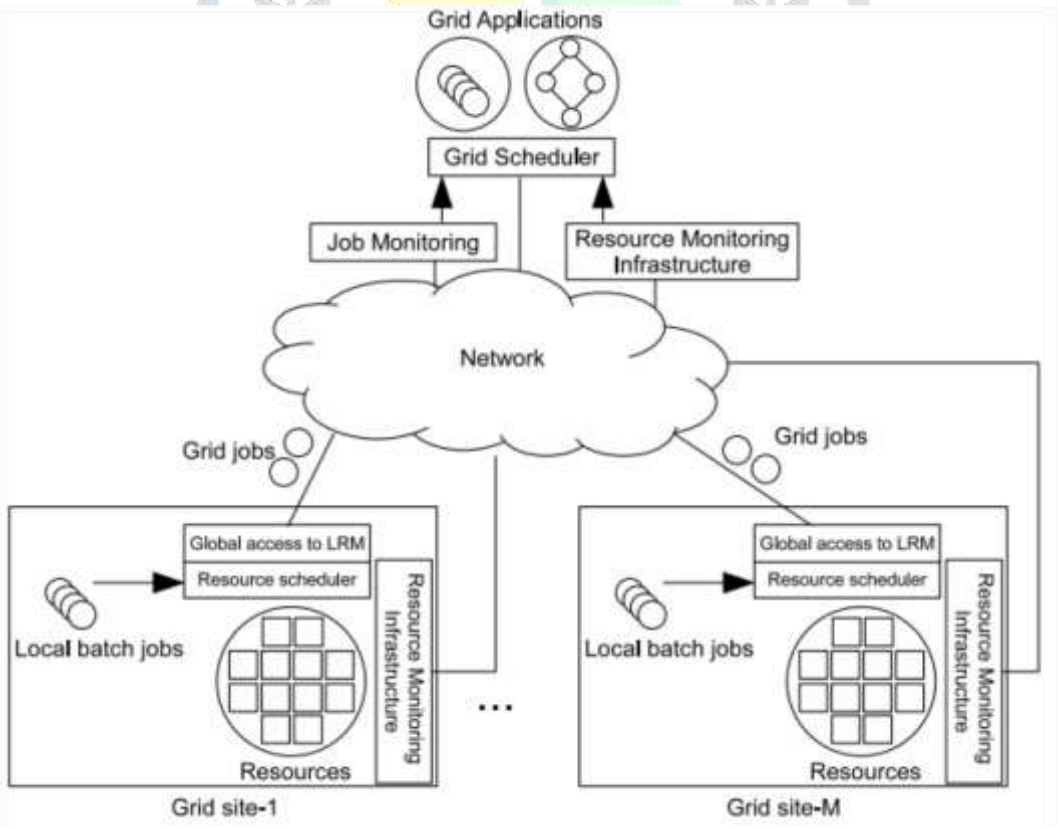

Figure 2: An architectural diagram showing core entities of Grid computing, In [1].

Scheduler get the information of available resources from GIS (Grid Information Service) and schedule the job according to it. The scheduling is affected by networking factor (such as bandwidth, distance etc.), scheduling approach (such as scheduling algorithm), resource availability, etc. The selection of resources [12] is also done by schedulers by following algorithm such as Max Heap, First Come First Serve, etc. The jobs which are obtained by local schedulers are allocated to the resources by GRID scheduler. GRID architecture is designed in that manner to meet the characteristics such as Security, Computation, Reliability, Versatile, etc.

Security: The user who want to process the jobs also wants to get his/her job in secure manner. The security is the function of the local machine which also follow some algorithm such as DES/3DES or TripleDES (Data Encryption Standard), Blowfish, etc.

Computation: The main characteristic is the computation through which the jobs are processed in very short time.

Versatile: The scheduler should be independent of the type of job, it should only concern with the processing of the jobs through which this term is facilitated.

The GRID architecture is divided into different layers such as Network layer, Resource layer, Middleware layer, Application layer [13].

- **Network layer:** The function of this layer to assure the connectivity for resources in grid. This also assure the connection in secure manner.
- **Resource layer:** This layer is only concern with the resources such as storage systems, computers, electronic data catalogues or sometimes even sensors, which are directly connected to the grid network.
- **Middleware layer:** This layer work as intermediator between various elements (servers, storage, networks, etc.) to provide them base to participate in unified grid environment to do so it uses various tools.
- **Application layer:** It consists of various application and tools for example- science, engineering, business, financial), also portal and development toolkits-supporting applications.

III. TYPES OF GRID COMPUTING

Grid has been divided into a number of types, on the basis of their uses:

3.1 Computational Grid

In [29], Computation are allowed in computational grid, optionally divided it into multiple parts and parallelly execute on different computers or grid nodes. Main benefit of computational grid is that computation will perform much faster and it can also use resources from all computers parallel. Map reduce is one of the most common design patterns for parallel execution. some of the computational grid features are:

Automatic Deployment: Only this feature provides largest productivity boosts in distributed system.

Collision Resolution: Gives the control to the user like which job get executed or which get rejected and how many jobs can be executed in parallel, which order to follow when they are executed.

Fail-over: In case of failure of jobs or node crush, grid jobs should automatically fail-over onto the other nodes.

Checkpoints: In case of fail-over it is very useful when a job is fail then failed job should be able to take its execution from latest point rather than the starting point.

3.2 Data Grid

When we need to distribute the data across the grid then we use data grid. The main aim of data grid is to provide as much as data possible from memory and to ensure data coherency. Some feature of data grid are as follows:

Data Replication: In [29], In the data grid all the data is replicated in all the nodes or computer in the grid. It is effective method for reading purpose. Whole data is available in all the nodes as immediate access.

Data Invalidation: In this feature, node loads the data on the other nodes on demand. Whenever data changes on the one node than data on the other nodes remain unchanged.so on the demand of nodes data is loaded on nodes the next time it is accessed.

Distributed Transactions: For data coherency transaction is required. Like database whenever we update the data or update failed, then the whole data must be rolled back. Most of the data grid support various type of transaction policies like write, read, serializable, etc.

Data partitioning: This feature of data grid permits the whole data divided into multiple subsets and assign every subset to every grid node. In this feature data is replicated between all the nodes and particular node is responsible for its own subset of data.

Data backups: whenever data is accidentally crushed then we need backup of the data. Data grid has ability to assign its data to other grid nodes. In this way crushed data can be recovered with the help of other nodes because data is available on the other nodes.

3.3 Network Grid

In [30], Network Grid provides high communication services between the network. Each grid node behaves as a data router between two communication nodes, providing data caching and high-speed communication facilities between the nodes. Work togather facility provide the grid network. Network grid is a type of grid topology in which each node is connected with the two or more nearer node along the one or more dimensions. If the chain of nodes has connected in the circular form along with one dimension then network topology named is ring topology. If the chain is connected in the circular form with along two or more dimension then the topology named is torus.

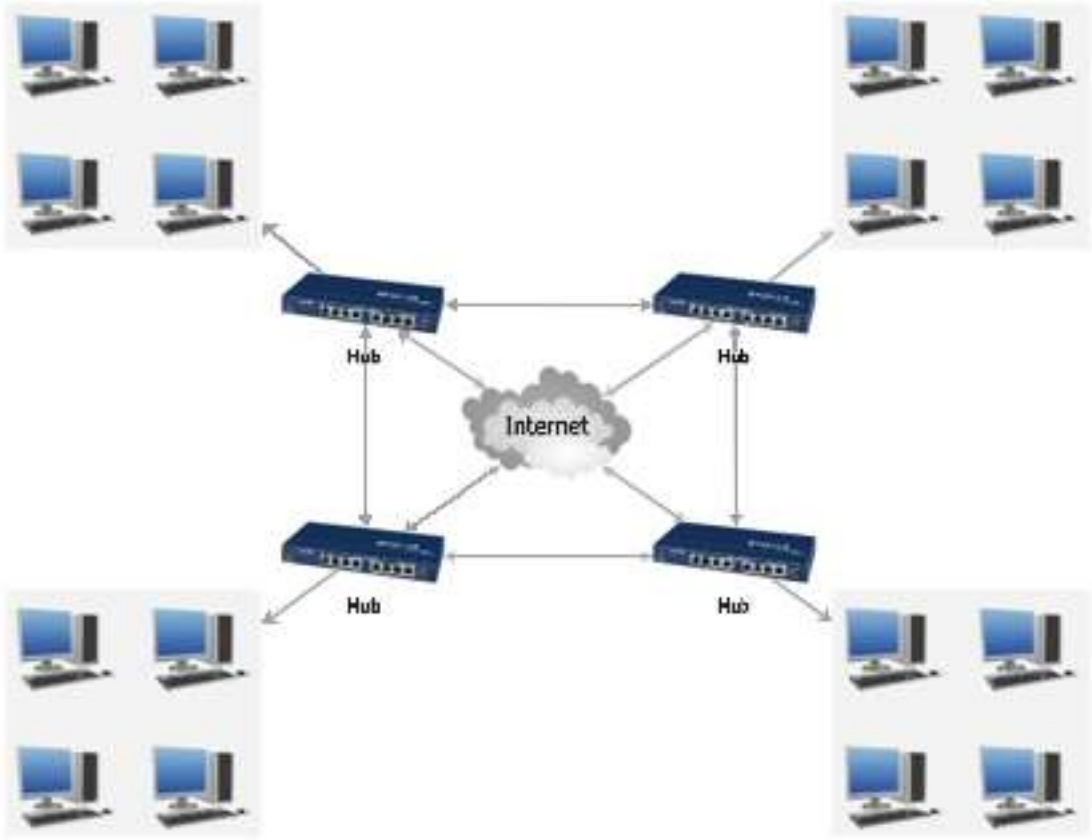

Figure 3: Grid network topology, In [17].

IV. SIMULATION TOOLS (GRIDSIM)

Simulation is used for operation of real world processes or systems. In [31,32], Simulation is actually running of a program that contains an algorithm or a mathematical equations.in various fields of computer science it is very useful. It is also very useful for modelling large complex system like parallel and distributed computing system. Simulation are very useful that permits the experimental without exposure risk. Simulation may be performed through:

4.1 Mathematical model

4.2 Game

Computer graphics model (such as flowchart of any program or equation or animated graph) etc. or many other purposes.

Advantages of Simulation toolkit like:

In [14], One of primary advantage of simulation is that there is no need to build real system. This allows the user to determine the efficiency and correctness of the design before system actually constructed [19].

User can perform the experimental works for a long time without limit [19].

There are many types of tools are available to simulate the jobs of grid, but it is mainly used for scheduling of grid jobs is Grid Sim.

Features of Grid Sim are as follows:

In [14], Grid sim is based on java toolkit for simulation. In [18], Static and Dynamic both schedulers are support in Grid Sim.Grid Sim can be used for dealing with other type of data grids. In [16], The Grid Sim toolkit permits modelling and simulation of distributed and parallel resourcemanagement, system users, applications and scheduling for grid environment. In [15], All segments In Grid Sim can communicate with each other through message cruising tasks characterized by Sim Java.It permits demonstrating of heterogeneous types of resources. Resources can be located at any time when required to be located. Multiple user can submit their tasks parallel for the same resources. Resources capability measured in terms of MIPS (Million Instructions Per Second). Grid sim is a powerful and advanced toolkit for simulation.

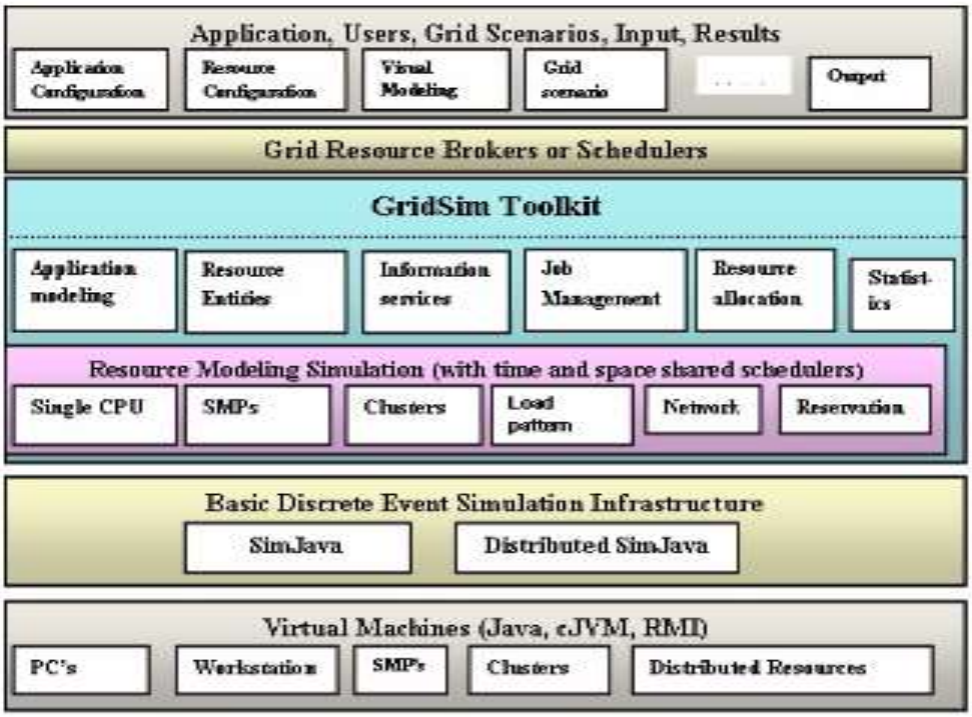

figure 4: Grid Sim Architecture, In [18]

V. **Applications of Grid Computing**

5.8 Distributed Supercomputing: In [23], An aggregate computational resource to dispose the problems which cannot be solved by a single system. To solve a very big or aggregate problem using single system it takes too much time. Using the distributed grid computing we can solve larger problems in very less time. For examples: climate modeling, computational chemistry etc.

5.2 High Throughput Computing: In [24], Clusters and Grids have been the favored stage for loosely coupled applications that have been generally part of the high throughput processing class of utilizations, which are determinate and executed through work process frameworks or parallel programming frameworks.

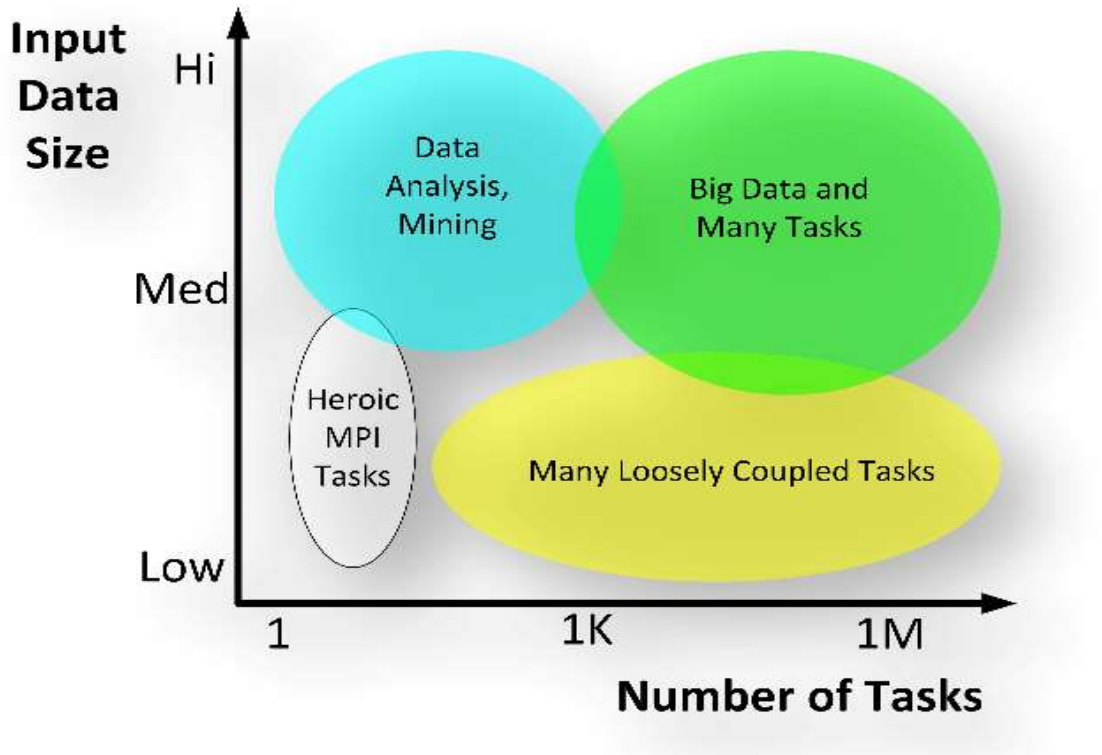

figure 5: Problems types with respect to data size and number of tasks, In [24].

5.3 On Demand Computing: On demand computing is the technology in which resources is In [21], provided whenever it is required. Some examples of on demand computing areas storage capacity, computational speed, and software applications.

5.4 Data Intensive Computing: In [22], A data intensive computing provides an environment that produce, manipulate or analyze data. A data-intensive application workflow has higher data overhead to manage than its computational parts. The interconnection of resources need to connected in a network for transferring data for the computational and processing task.

5.5 In [26], To solve real life problems scientists and engineers are developing more and more complex applications and large data sets, and process scientific experiments on distributed resources.

5.6 Collaborative Computing: - A collaborative computing is empowered shared use of data backlog and simulations. In [23], Virtual collaboration is the utilization of computerized advancements that empower associations or people to cooperatively design, outline, create, manage and research items, services and imaginative applications.

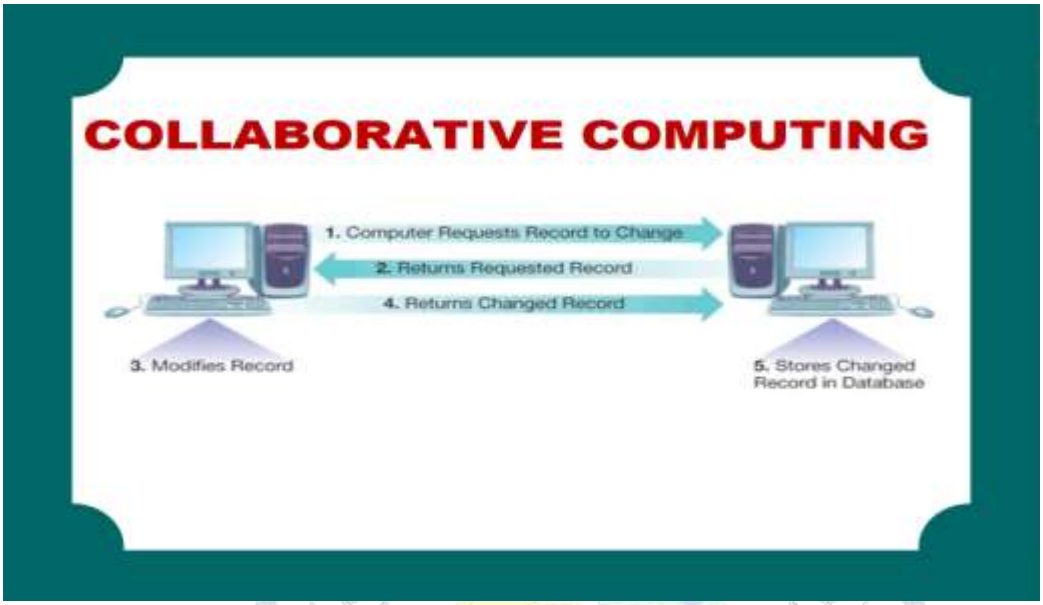

Figure 2: Collaborative Computing, In [25].

5.7 In [27], The grid computing or computational provides large-scale resource sharing, for example personal computer, clusters, Data base and online instructions.

5.8 In [28], Grid computing technology has been applied to computationally intensive mathematical and academic problems through volunteer computing, and it is used in commercial enterprise for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back office data tasking in support for e-commerce and web services.

VI. Advantages of Grid Computing

- No need to buy mainframe or super computer or server for execution of the large number of process.
- Utilization of idle resources.
- In grid computing jobs can be processed in parallel therefore it increases the performance and throughput or reduce the execution cost.
- Due to modular environment of grid, it decreases the points of failure.
- A grid software easily managed the all policies of grid.
- Grid computing focus on the quality of the services.
- Relatively low cost as compare to supercomputer, mainframe servers.
- It is useful for executing large number of jobs.
- Grid computing can increase the computational capacity of the company (or organizations).
- It provides worldwide access resources throughout the distributed network.
- Enable Co-partnership for virtual company.
- Adaptable, Secure, Coordinated resource sharing.
- Increment accessibility of uses while decreasing both infrastructure and management costs.
- Grid computing can solve larger complex problem in shorter time.

- In [26], Grid computing having ability to build dynamic application of the distributed resources.

REFERENCES

[1] Harshadkumar B. Prajapati, Vipul A. Shah. 2014. Scheduling in Grid Computing Environment.

[2] Fourth International Conference on Advanced Computing & Communication Technologies.

[3] Seema Kumari, Gulshan Kumar, April 2015, "Survey on Job Scheduling Algorithms in Grid Computing", International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 15.

[4] Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi. Dec. 2005. Introduction to Grid Computing. IBM International Technical Support Organization.

[5] Mark Baker, RajkumarBuyya, Domenico Laforenza. 2002. Grids and Grid technologies for wide-area distributed computing. Softw. Pract. Exper. 2002; (in press) (DOI: 10.1002/spe.488).

[6] Manjeet Singh. 2016. Incremental Checkpoint Based Failure-Aware Scheduling Algorithm in Grid Computing. International Conference on Computing, Communication and Automation (ICCCA2016), ISBN: 978-1-5090-1666-2/16/$31.00 ©2016 IEEE.

[7] Minakshi Memoria, Dr. Mukesh Yadav. March 2016. Fault Tolerance in Grid Computing with Improved Resource Utilization and Maximum Efficiency. International Journal of Engineering Science and Computing.

[8] S. JagadesanMCA.,M.Phil.,M.E., M.Vanitha. February 2016. An Adaptive Counting Job Scheduling algorithm for grid computing. International Journal of Computer Trends and Technology (IJCTT) – Volume 32 Number 1.

[9] Sunita Bansal, Bhavik Kothari, ChittaranjanHota. March 2011. Dynamic Task-Scheduling in Grid Computing using Prioritized Round Robin Algorithm. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2.

[10] Vishnu Kant Soni, Raksha Sharma, Manoj Kumar Mishra. 2010. Grouping-Based Job Scheduling Model In Grid Computing. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:4, No:5.

[11] K JairamNaik, Dr A Jagan, Dr N Satya Narayana. 2015. A novel algorithm for fault tolerant job Scheduling and load balancing in grid computing environment. 2015 Inteational Conference on Green Computing and Internet of Things (ICGCloT), 978-1-4673-7910-6/15/$31.00 ©2015 IEEE.

[12] Mandeep Kaur. 22 Jan - 24 Jan 2016. FastPGA based Scheduling of Dependent Tasks in Grid Computing to provide QoS to Grid users. International Conference on Internet of Things and Applications (IOTA) Maharashtra Institute of Technology, Pune, India.

[13] S. Gomathi, Dr.D.Manimegalai. 2011. An adaptive grouping based Job Scheduling in Grid Computing. Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011), 978-1-61284-653-8/11/$26.00 ©2011 IEEE.

[14] Prof. Robert van Engelen. 2008. Concepts and Architecture of Grid Computing, HPC II Spring.

[15] C.Babitha Lilly Mary, R.Rajagopal, Dr.M. Chitra. March 2014. Job Scheduling and Schedulers in Grid Computing- A Survey, ISSN 2278-1447.

[16] RajkumarBuyya. GridSim: A Grid Simulation Toolkit for Resource Modelling and Application Scheduling for Parallel and Distributed Computing.

[17] Anthony Sulistio, UrosCibej, SrikumarVenugopal, BorutRobic,RajkumarBuyya. 2002/09/19. A toolkit for modelling and simulating Data Grids: An extension to GridSim.

[18] http://www.conceptdraw.com/How-To-Guide/grid-network-topology

[19] Mrs. R. Kingsy Grace, S. SwathiPriya, S. Surya. December 2012. A Survey on Grid Simulators. IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 2, No.6.

[20] Arnaud Legrand, Loris Marchal, Henri Casanova. 2003. Scheduling Distributed Applications: the SimGrid Simulation Framework. Third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03), pp.138.

[21] William H. Bell, David G. Cameron, Luigi Capozza,A. Paul Millar, Kurt Stockinger, FlorianoZini. 2003. OptorSim-A Grid Simulator for Studying Dynamic Data Replication Strategies. International Journal of High Performance Computing Applications, Vol.17 (4), pp-403-416.

[22] On demand computing techopedia (https://www.techopedia.com/definition/1308/on-demand-computing-odc).

[23] Suraj Pandey, RajkumarBuyya. 2009. A Survey of Scheduling and Management Techniques for Data-Intensive Application Workflows. Information Science Reference (an imprint of IGI Global) USA.

[24] Nagaraj Baddi. 2009-10. Grid Computing. A seminar report (2SD07CS402), Sri DharmasthalaManjunatheshwaraCOLLEGE OF ENGINEERING AND TECHNOLOGY.

[25] IOAN RAICU. March 2009. MANY-TASK COMPUTING: BRIDGING THE GAP BETWEEN HIGH-THROUGHPUT COMPUTING AND HIGH-PERFORMANCE COMPUTING, THE UNIVERSITY OF CHICAGO.

[26] Baldric Horton, Slide player, TELECOMMUNICATION & NETWORK APPLICATIONS PART 2 (http://slideplayer.com/slide/7109078/)

[27] Jia Yu,RajkumarBuyya. 2006. A Taxonomy of Workflow Management Systems for Grid Computing.

[28] Pinky Rosemarry, Ravinder Singh, Payal Singhal and Dilip Sisodia. December 2012. GROUPING BASED JOB SCHEDULING ALGORITHM USING PRIORITY QUEUE AND HYBRID ALGORITHM IN GRID COMPUTING, (IJGCA) Vol.3, No.4.

[29] Dr. T. Kokilavani. 2017. Enhanced Round Robin Technique with Variant Time Quantum for Task Scheduling InGrid Computing, IJETST- Vol.||04||Issue||09||Pages 6016-6021||September||ISSN 2348-9480.

[30] Compute Grids vs. Data Grids,https://dzone.com/articles/compute-grids-vs-data-grids

[31] The Grid Computing Weblog,http://thegridweblog.blogspot.in/

[32] Computer simulation-Wikipedia, https://en.wikipedia.org/wiki/Computer_simulation/