International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 1, January-February 2015 **ISSN 2278-6856**

Performance evaluation on a Grid platform

Asif Ali Banka[1] Muneer Ahmad Dedmari[2], Mufti Tufail Masood[3]

[1].Deptt. Of Computer Science & Engineering Islamic University of Science & Technology
Awantipora, J&K India

[2,3] Deptt. Of Computer Science & Engineering Islamic University of Science & Technology
Awantipora, J&K India

Abstract

The technological advancements have resulted in substantial increase of commodity computing, mainly as the outcome of faster hardware and more sophisticated softwares. In spite the presence of super computers in present age, all the problems in the fields of science, engineering and business cannot be efficiently and effectively dealt with. This is mainly because of complexity factor and cost margin. For a complex program presented by any of the fields above, the data provided requires a number of heterogeneous resources that are scattered across the globe hence making the problem very cumbersome to handle. To address this; concept of grid computing evolved which combines and connects all the required heterogeneous resources to form a single entity which can resolve the problem at hand. With the implementation of grid at practical level lot of problems arise which need to be addressed before the system is put to use. One of the main aspects that must be kept under check while executing a problem set on grid is the security, which mainly includes the privacy and the integrity of the data that becomes vulnerable due to its distributed nature. This paper mainly focuses on the implementation of grid and gives the idea of the exponential difference between the performances of a stand-alone system in comparison to grid. In its broader view we intend to discuss analysis of a specific problem set on both the platforms, and provide the analysed data to support the high end performing nature of the grid in contrast to a stand-alone system.

Index Terms:-Grid computing, grid components, grid setup, performance evaluation.

1. INTRODUCTION

Grid Computing can be considered as an infrastructure service that makes computation available on demand like water, gas or electricity by joining resources spread over the globe at different locations. Grid is virtualization of resources which is a hardware software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to computational capabilities, as described in Ian Foster and Carl Kesselman [1]. In grid computing sharing of computational resources (CPU cycles, disk space etc.) rather than data is be attained. Its key values are in the underlying distributed computing infrastructure technologies that evolve in support of cross-organizational application and resource sharing (virtualization) across platforms and organizations. This kind of virtualization is achievable through use of open standards, which ensure that applications can transparently take advantage of whatever appropriate resources made available to them. Grid computing could be defined as any of a variety of levels of virtualization along a continuum. Exactly where along that continuum one might say that a particular solution is an implementation of grid computing versus relatively simple implementation using virtual resources is a matter of opinion [2].

In most organizations there are lots of underutilized resources, such as computing and data resources. With the help of grid these unused resources can be properly utilized. Often, resources may have enormous unused disk capacity. Grid computing (more specifically, a data grid) can be used to aggregate the unused storage into a larger virtual data resource.

Grid also offers resource balancing by scheduling jobs on machine with low utilization.

On the basis of use, grid computing can be divided into different types:

- **Computational grids:** These type of grid are meant to provide secure access to computational resources, sufficient enough to perform processing of computational problems which otherwise would have required high computing power machines.
- **Collaboration grid:** With the advances in network hardware resources and Internet services, demand for better collaboration has increased. Such desired collaboration is best possible with these kinds of grids.
- **Utility Grid:** In this type of grid not only CPU cycles are shared, also other software's and special peripherals like sensors are also shared.
- **Network grid:** Even if we have computational machines with enough computational power as a part of grid but with poor network communication one can't utilize those machines optimally. Network grid provides high performance communication using data caching between nodes there by speed-up communication with each cache nodes acting as router.
- **Data grid:** There are two things, data and computation over that data. Data grid provides the support for data storage other data related services like data discovery, handling, publication, etc [3].

Globus: A software infrastructure that enables applications to handle distributed, heterogeneous computing resources as a single virtual machine. The Globus project is a U.S. multi-institutional research effort that seeks to enable the construction of computational grids. A computational grid, in this context, is a hardware and software infrastructure that provides dependable, consistent, and pervasive access to high-end computational capabilities, despite the geographical distribution of both resources and users. Globus is constructed as a layered architecture in which high-level global services are built upon essential low-level core local services. The Globus toolkit is modular, and an application can exploit Globus features, such as resource management or information infrastructure, without using the Globus communication libraries [4].

SETI@HOME: The seminal Internet distributed computing project, SETI@home, originated at the University of California at Berkeley. SETI stands for the "Search for Extra-terrestrial Intelligence," and the project's focus is to search for radio signal fluctuations that may indicate a sign of intelligent life from space. SETI@home is the largest, most successful Internet distributed computing project to date. Launched in May 1999 to search through signals collected by the Arecibo Radio Telescope in Puerto Rico (the world's largest radio telescope) [5].

2. IMPLEMENTATION

For realizing grid following steps should be taken into consideration:

1 Integration and communication of various individual software and hardware components to act like a single system.

2 Implementation of middle-ware so that the resource sharing and access becomes secure.

3 Development of grid tools so that grid applications and resources can be easily monitored and managed.

4 Developing applications that can be used to take the advantages of resources available in grid [6].

The components required to establish a grid are:

Grid Fabric: It includes geographically distributed and accessible from anywhere on the Internet resources. It comprises computers (workstations or PC's running on different operating systems), databases, networks, resource management systems such as LSF, CONDOR or PBS and sensors or special scientific devices.

Grid Middle-ware: It acts as intermediator between components of software with the applications, allowing them to communicate. It is made up of services sets such as remote process management, co-allocation of resources, information exchange, and security. Interoperability between two or more networked computers is enabled by grid middleware with the help of common protocols.

Grid Tools: These tools helps in realizing grid, developing grid enabled applications and brokers that acts as user agents which can manage or schedule computational resources across grid.

Grid Applications and Portals: Grid portals provide us user interface for accessing grid services. It can be used for submission of jobs and getting result back in grid environment [7].

While creating the grid environment, we walked through the following processes/tasks

1) Globus Toolkit Configuration: The Globus Toolkit is an open source software toolkit used for building computational grids and grid based applications. It allows sharing of computing power, databases, and other tools securely online across corporate, institutional and geographic boundaries without sacrificing local autonomy. It is being developed by the Globus Alliance. It is widely used as a basis for grid implementations all over the world, including Grid Ireland. The Globus toolkit has a modular design which contains at its core an implementation of the Open Grid Services Infrastructure (OGSI) specification. Globus provides resource management, data management and information services. Built upon this, there is a security infrastructure which provides security at a message level as well as authentication and authorization [8].

2) Network Setup: Configuring DNS: Domain Name System (DNS) converts the name of website (www.gridboss.mquad.com) to an IP address (192.168.0.6). This is important, because the IP address of a Web site's server, not the Web site's name, is used in routing traffic over the Internet. Everyone in the world has a first name and a last, or family, name. The same thing is true in the DNS world: A family of Web sites can be loosely described a domain. For example, the domain linuxhomenetworking.com has a number of children, such as www.linuxhomenetworking.com and mail.linuxhomenetworking.com for the Web and mail servers, respectively. The DNS queries from remote machines can be responded by BIND. It is an acronym for the Berkeley Internet Name Domain project, which is a group that maintains the DNS-related software suite that runs under Linux [9].

3) Security Implementation: For implementing security in grid environment one must consider the following security aspects.

- Authentication: Is the process by which a resource or user's identity is verified. It is extremely important to correctly establish the correct identity of an entity.

- Authorization: Once the identity of an entity is correctly established authorization is used to find what actions the entity is entitled to perform.
- Encryption: Any data passing over public networks is open to be intercepted. A sniffer has the ability to listen in on this data and extract sensitive information as well as usernames and passwords. To avoid this encryption is used to encrypt all transmissions. To a party sniffing the data it would appear as garbage.
- Data Integrity: This involves ensuring that the integrity of the data transferred across the Grid is maintained. This is extremely important in guaranteeing accurate results from jobs.
- Data Confidentiality: A lot of grid applications involve processing data. In many cases where this data is of a sensitive nature it is imperative to ensure that only those authorized to view this data be allowed.
- Single Sign On and Delegation: It is often the case that an entity A will act on behalf of entity B and will need to contact another entity C on B's behalf. Delegation of entities' credentials enables this to be possible [10, 11].

The attempt to fulfill these security requirements systems based on the Public Key Infrastructure (PKI) were used. The Globus Toolkit offers a Grid Security Infrastructure (GSI) that provides a PKI based security system that can be used to address most of the security problems encountered when dealing with grids [12].

For the Globus toolkit, X509 certificates were used for authenticating users, hosts and resources. This package uses OpenSSL commands to create public and private keys for the Certifying Authority (CA) [11].

3. RESULT

The implemented grid system is analyzed to see how it performs in contrast with stand-alone system. The performance of both the systems can be analyzed by the time taken for a job (operation) to get completed. Difference between the times taken by two systems to complete a job can be evaluated by:

$$t_d = t_s - t_g \qquad (1)$$

where t_s is the time taken by stand-alone system to complete the job, t_g is the time taken by grid system to complete the job, and $t_{d\ is}$ the difference of time taken by the grid system and the stand-alone system to get job completed.

The jobs (operations) this paper is focused on are:

Copy operation: In this operation different set of data of varying size has been copied from one location to another on the same system both by the grid and stand-alone system. As shown in Fig1. the time taken by a stand-alone system for carrying out the copy operation is more than that taken by the grid system for the same job because in grid, computational power of different systems are utilized for carrying out the operation.

TABLE.1 REPRESENTS THE TIME TAKEN BY GRID AND STAND ALONE SYSTEM TO COPY THE DATA OF DIFFERENT SIZES.

Time Taken By A Stand Alone System(In Sec)(Ts)	Time Taken In Grid(In Sec)(Tg)	Size(In GB)
0.86	0.79	0.1457
1.452	1.252	0.1958
13.236	7.134	0.3730
14.121	10.975	0.5689
27.049	25.59	0.746
31.023	28.263	0.7649
41.999	45.649	1
50.979	47.32	1.4
92.596	86.081	2
135.42	129.26	3
180.13	174.46	4
220.56	201.16	4.8
355.12	345.79	8
1661.68	1499.36	15.5
3265.68	2956.66	32

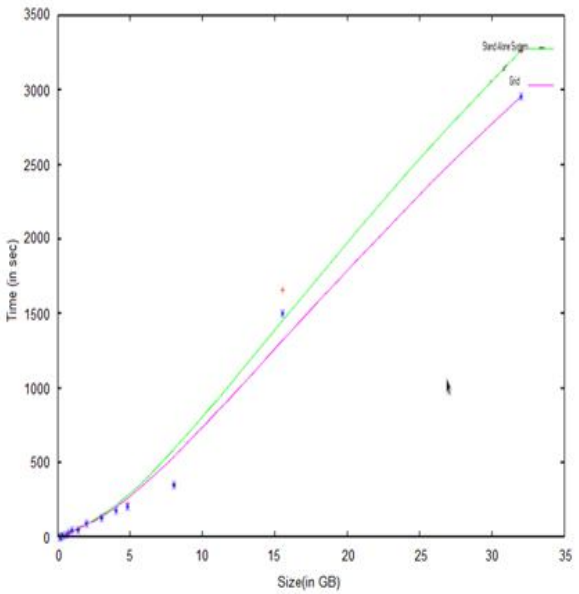

Fig1.Copy operation submitted to stand alone system and

From graph shown on Fig 1, it is evident that the copy operation processed on grid takes less time when compared with a standalone system.

ii) Tar operation: It is concerned with the archiving of different files into one compressed entity. On grid these different files are compressed using resources of multiple components so as to enhance the overall performance for carrying out the mentioned operation.

Table.2 represents the time taken by grid and stand-alone

Time Taken By A Stand-Alone System(In Sec)(Ts)	Time Taken In Grid(In Sec)(Tg)	Size(In GB)
0.032	0.028	0.0079
1.448	0.475	0.1457
1.754	1.542	0.1958
7.542	7.139	0.3730
12.045	11.477	0.5689
27.233	24.212	0.7649
38.145	35.968	1
57.19	50.15	1.4
100.011	95.79	2
147.56	140.67	3
192.16	182.76	4
236.06	217.56	4.8
370.72	352.16	8
1168.21	989.32	15.5
2502	1573	32

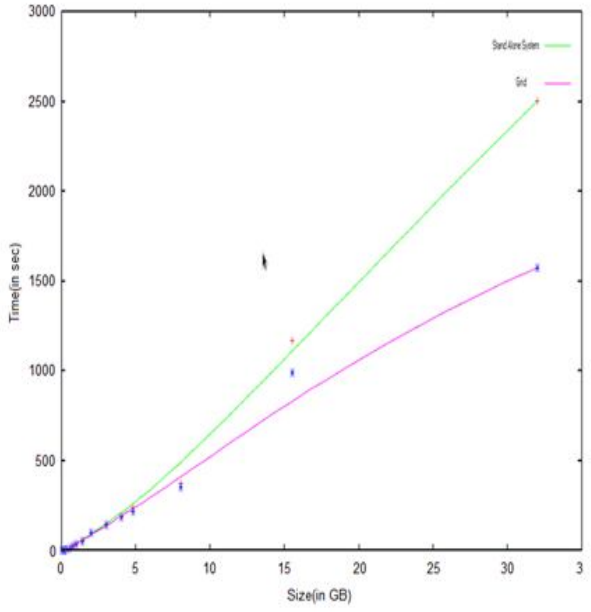

Fig.2 Tar operation submitted to stand-alone system and grid.

Fig 2 clearly signifies that performance of operation on the stand-alone system is less than the same operation performed on the grid.

iii) Untar Operation: It deals with expansion of a compressed file back into its original form. The decompression is done over different machines by the distribution of the compressed data on the grid and then collected to form the original. The comparison of performing the operation on both the platforms reviewed that the Untar operation on the stand-alone system took more time than the same operation done on the same file on the grid.

TABLE.3 REPRESENTS THE TIME TAKEN BY GRID AND STAND-ALONE SYSTEM FOR DOING UNTAR OPERATION ON DIFFERENT SIZES OF DATA.

Time Taken By A Stand-Alone System(In Sec)(Ts)	Time Taken In Grid(In Sec)(Tg)	Size(In GB)
0.046	0.027	0.0079
1.855	0.510	0.1457
2.301	1.627	0.1958
8.403	7.564	0.3730
29.56	27.52	0.5689
38.145	34.331	0.7649
47.230	42.778	1
65.45	58.65	1.4
107.517	99.70	2
158.77	150.76	3
199.79	187.66	4
249.24	238.56	4.8
395.96	360.23	8
1405.11	1107.56	15.5
2939	1893	32

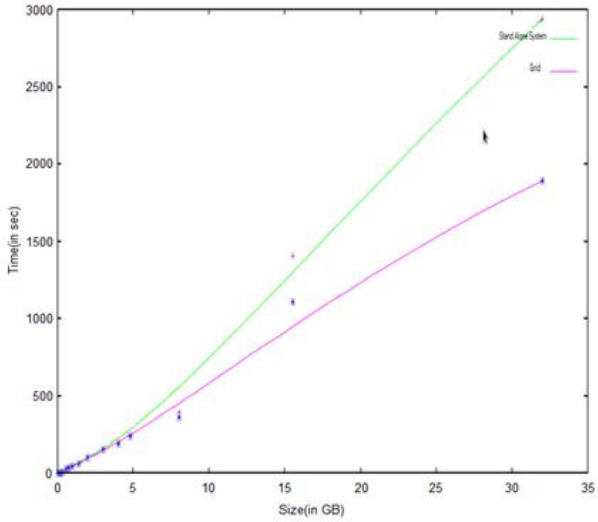

Fig 3 .Untar operation submitted to stand-alone system and grid

In consideration to Fig 3, it is consistent that grid performs faster than a standalone system.

When compared, the results of operations done on both the system, it is concluded that grid does no doubt perform a specific task much faster than the same operation performed on the stand alone system. These operations being COPY, TAR and UNTAR.

This can be verified by calculating the average time taken by the stand alone system and the grid system in TAR, UNTAR and COPY operations individually (The average time taken by grid (t_g) being 6091.905 sec, 4862.11 sec, 5648.377 sec for copy, tar and untar operations respectively and the time taken by stand alone system (t_s) being 5559.744 sec, 3682.251 sec, 4210.477 sec for copy, tar and untar operations respectively). Substituting the values of each in equation I:

COPY operation:

$$t_d = 6091.905 - 5559.744 = 532.161\text{sec}$$

TAR operation:

t_d = 4862.11 – 3682.251 = 1179.859sec

UNTAR operation:

t_d = 5648.377 – 4210.477 = 1437.9 sec

The positive values of 't_d' for each operation above show that grid actually performs better than the stand alone system.

Furthermore, performance of grid system is dependent on certain factors and this can be shown by:

$$t_p \propto \frac{S}{\sum_{i=1}^{n} Pi} \quad (2)$$

were t_p is time taken by grid system to complete an operation, S is size of the archiving operation and P_1, P_2, P_3 ..... P_n are the available processing power of n nodes participating in grid system.

It can also been seen that the variance in the time taken for the TAR and UNTAR operations in comparison with the stand-alone system is much larger than the variance observed in case of the COPY operation.

4. CONCLUSION

With each passing nano second the complexity of problems provided to the stand alone computers is increasing exponentially so for now the preexisting super computers focused on a specific computational method may handle the problem but in the near future the one stand-alone super computer may not be sufficient to solve the high complexity problems. As we have presented in this paper, in support by the results that the factor by which the performance of computation on a problem set by utilizing the various resources around the globe is actually very high as compared to the same computation being performed on the stand alone computer. The fact that a technology can connect all the specific resources around a globe required for a problem and then single handedly solve the same by a high performance factor and in an affordable manner as that of a super computer of present or future gives us the glimpse of the importance of grid computing.

ACKNOWLEDGMENT

The work of course would not have been completed without support from family and guidance provided by the supervisor. It is worth mention that Mr. Naiman Altaf and Mr. Mohammad Nayeem Shah has provided support and vital role in completion of this research paper.

REFERENCES

[1] Ian Foster, Carl Kesselman, "The GRID: Blueprint for a new Computing Infrastructure," Morgan Kauffman Publishers, 1999.

[2] Bart Jacob, Michael Brown, Kentaro Fukui and Nihar Trivedi, "Introduction to grid computing", listed in IBM RedBooks, 2005,pp.03–04.

[3] Ovaise Khan, "Types of Grid", http://thegridweblog.blogspot.in/2005/10/types-of-grids.html.

[4] Luis Ferreira, ViktorsBerstis, Jonathan Armstrong, Mike Kendzierski, Andreas Neukoetter, Masanobu Takagi, Richard Bing-Wo, Adeeb Amir, Ryo Murakawa, Olegario Hernandez, James Magowan, Norbert Bieberstein, "Introduction to Grid Computing with Globus", IBM RedBook.

[5] SETI@home: Search for Extraterrestrial Intelligence at home, "http://setiathome.ssl.berkeley.edu/"

[6] Mark Baker, Rajkumar Buyya and Domenico Laforenza Grid, "International Efforts in Global Computing"

[7] JC Desplat, Judy Hardy, Mario Antonio letti, Jarek Nabrzyski, Maciej Stroinski, Norbert Meyer, "Grid Service Requirements", Enacts, 2002

[8] Ian Foster and Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", International Journal of Supercomputer Applications, 11(2): 115-128, 1997

[9] Paul Albitz and Cricket Liu, "DNS and BIND", 4TH edition, 2001, pp. 04-05.

[10] Ian Foster, Carl Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids", 5th ACM Conference on Computer and Communications Security Conference, San Francisco, CA, USA.

[11] S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile", http://www.hjp.at/doc/rfc/rfc3820.html

[12] Shushan Zhao, A. Aggarwal, R.D. Kent, "PKI-Based Authentication Mechanisms in Grid Systems", IEEE Conference on Network, Architecture and Storage, 2007.