International Journal of Research in Advent Technology (IJRAT) (E-ISSN: 2321-9637)
Special Issue
National Conference "CONVERGENCE 2016", 06th-07th April 2016

Virtual Math Server Using Grid Computing

Sujit R.Borey[1], Dr.P.M Jawandhiya[2]

M.E.CSE PLIT [1] ,PRINCIPAL PLIT [2]
Email:sujeetbore3@gmail .com[1] , principal@plit.ac.in[2]

Abstract-This Grid computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. Grid technologies promise to change the way organizations tackle complex computational problems. Grid Computing got its name because it strives for an ideal scenario in which the CPU cycles and storage of millions of systems across a worldwide network function as a flexible, readily accessible pool that could be harnessed by anyone who needs it, similar to the way power companies and their users share the electrical grid.

In this project we are interested to show the potential use of Grid Computing in small organization to reduce the time required to perform particular task & use the resources efficiently. We are developing the **'3-tier Architecture'** Distributed Network that is intend to integrate various Mathematical Applications Virtually. In first tier we are developing the User side Interface. In this we are accepting the Computing task from User i.e. a mathematical expression. Second tier is very crucial part of this Project, in Grid Terminology we can say it a Middleware or MidServer Node. This receives the task from User, redirects it to appropriate workstation, according to Operation over the Network, collects the result, return it to User. Third tier comprises the End Servers which participates by doing actual computing. These servers are doing specific Task. For example Addition, Multiplicationetc. These will receive the Task from Master, Computes the result & returns the Result.

Index Terms Grid Computing, MidServer, Distributed Network, Network computing **.**

1. INTRODUCTION

1. Distributed computing

Distributed computing is a field of computer science that studies distributed systems. A distributed system consists of multiple computers that communicate through a computer network. The computers interact with each other in order to achieve a common goal. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs.

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.[2]

2. Grid Computing

Grids enable the sharing, selection, and aggregation of a wide variety of resources including supercomputers, storage systems, data sources, and specialized devices that are geographically distributed and owned by different organizations for solving large-scale computational and data intensive problems in science, engineering, and commerce. Thus creating virtual organizations and enterprises as a temporary alliance of enterprises or organizations that come together to share resources and skills, core competencies, or resources in order to better respond to business opportunities or large-scale application processing requirements, and whose cooperation is supported by computer networks.[3] The concept of Grid computing started as a project to link geographically dispersed supercomputers, but now it has grown far beyond its original intent. The

Grid infrastructure can benefit many applications, including collaborative engineering, distributed supercomputing. [I][9]

- **International**
  - Commercial
    - Microsoft,IBM,HP
  - Non-Commercial
    - BOINC, Globus
- **National**
  - Commercial
    - Data not Available
  - Non-Commercial
    - C-DAC's GARUDA1.3

3. Drawbacks of Existing System: The existing system imposes a lot of overhead on the users. The primary drawbacks of the existing system are

- Huge wastage of CPU cycles
- Memory wastage
- Bandwidth Bottleneck
- Deals with homogeneous dataset

4. Proposed System

This project aims to solve the mathematical expression of user submitted task with form of string. This project mainly aims to view the performance of the computers in a network. This concept is achieved by using (socket and servlet programming) java. The main objective of this project is to get the Remote system configuration, the remote system current running processes, CPU idle time and the total memory of the remote system (i.e. free and usage). So when a task come in it will be processed immediately based on the performance analysis details, thereby increasing the processing speed and time efficiency[II][1].

In SETI@HOME project researchers hope to gain as much insight as possible into the existence of extra-terrestrial intelligence. Another trend associated with high-performance computing is Internet computing where Web services can be provided to address the needs of clients.[10]

The proposed system is to develop a "Grid Technology". The proposed system has the following objectives to solve the problems in the existing system.

- To get the Remote system configuration, CPU idle time and the total memory of the remote system (i.e. free and usage).
- To exploit the inherent distributed nature of an application.
- To decrease the turnaround/response time of a huge application.
- To allow the execution of an application this is outside the capabilities of a single (sequential or parallel) architecture.
- This project has three tier Architecture and it is LAN based application.

  Its perfoming four basic arithmatic operations that are addition, subtraction, multiplication, division. These operation are running on End Servers which are collaberativelly working with MidServer.

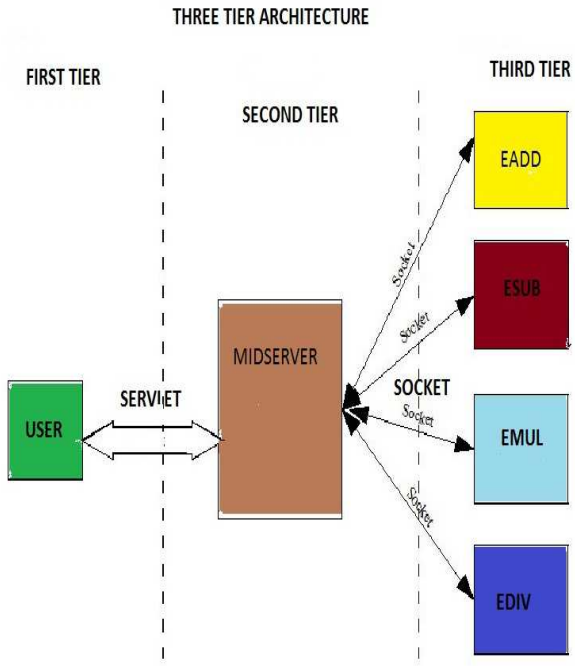

Fig.1.Architecture of project.

5. Parallel and distributed computing

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:[4][10]

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.[5]

The figure on the right illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure

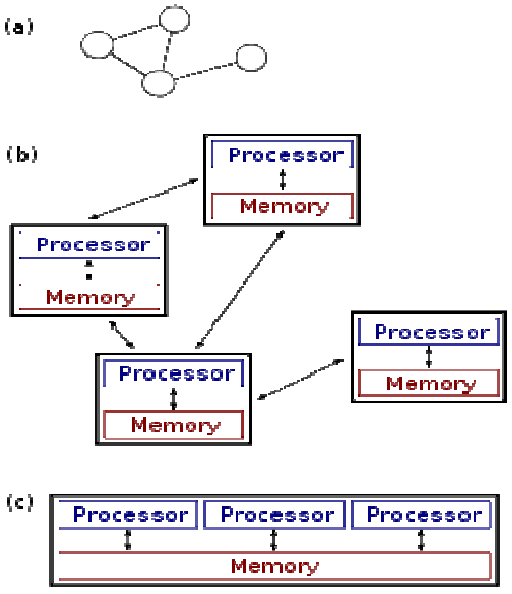

Fig. 2 (a)–(b) A distributed system (c) A parallel system.

(b) shows the same distributed system in more detail: each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.[7]

The situation is further complicated by the traditional uses of the terms parallel and distributed *algorithm* that do not quite match the above definitions of parallel and distributed *systems*; see the section Theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.[6]

6. Applications

There are two main reasons for using distributed systems and distributed computing. First, the very nature of the application may *require* the use of a communication network that connects several computers. For example, data is produced in one physical location and it is needed in another location.

Second, there are many cases in which the use of a single computer would be possible in principle, but the use of a distributed system is *beneficial* for practical reasons. For example, it may be more cost-efficient to obtain the desired level of performance by using a cluster of several low-end computers, in comparison with a single high-end computer. Examples of distributed systems and applications of distributed computing include the following:[8]

Network applications:

- World Wide Web and peer-to-peer networks.
- Massively multiplayer online games and virtual reality communities.
- Distributed databases and distributed database management systems.

❖ **Real-time process control:**

- Aircraft control systems.
- Industrial control systems.

❖ **Parallel computation:**

- Scientific computing, including cluster computing and grid computing and various volunteer computing projects; see the list of distributed computing projects.
- Distributed rendering in computer graphics.

CONCLUSION:

Now a day's millions of PCs are connected in various ways. These networked computers are not well utilized up to their capability. These unutilized computers can be used well and optimally using the Grid computing. Using this technology we don't need super computers. In research organizations like ISRO, IBM uses super computers. These super computers are more costly comparing with Personnel computers. But these

Networked personnel computers can be well utilized and can be perform tasks that are commonly given to the super computers. This is possible because of the Millions of computers on the Internet can be taped. The implementation of the Grid Computing is difficult to program but once we develop the application it can be used to for the tenure. But the only pitfall is that all applications and all tasks are to be reprogrammed so that they can work with this Grid computing technique. This is more expensive work. If the heavy task programs that are currently being developed are implemented with Grid computing, then all computers will automatically well utilized via the network, provided that all computers must give security privileges to each other.[8]

This Project is developing five Grid features:

1. Load distribution

Load distribution is a computer networking method to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load distribution, instead of a single component, may increase reliability through redundancy.

2. Virtual Integrity

All the method invocations behave as if they were local computations. In particular, the client can ignore any question of where the service is located (this is oftentimes referred as *location transparent*)[3].

3. Increase in Computation capability

A problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

4. Efficient use of Computing Resources

Allows applications to use grids to put unused processor cycles to work in generally loosely coupled or independent tasks

5. Multi User

Multiple users can run their applications on the server simultaneously.

FUTURE SCOPE:

In the future this application has the more scope to incorporate more complex operations like Digital Image processing, Pattern Matching, Computation of Complex numbers & so on. We can do it by using more sophisticated tools for example CORBA & RMI in Java. In forthcoming years we can have many network enabled devices. So by using this kind of mechanism we utilize them up to their maximum capacity.

REFRENCES

I. Ahmar Abbas, Grid Computing: Practical Guide to Technology & Applications| ISBN: 1584502762 | 2003 | CHM | 406 page

II. Elliote Rusty Harrold, Java Network Programming | ISBN- 10: 0596007213 | 2009 | O'Reilly Media | 764 page

Bibliography:

1. Andreas Savva, Toshiyuki Suzuki, HiroKishimoto, "Business Grid Computing Project Activities", **FUJITSU Scientific and Technical Journal**, 40, 2, p. 252-260, 2004.
2. I.Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration". Open Grid Service Infrastructure WG**, Global Grid Forum**, June 22, 2002
3. David P. Anderson, Je_ Cobb, Eric Korpela, Matt Lebofsky, and DanWerthimer. Seti@home: an

experiment in public-resource computing**.** ***Communications of the ACM***, 45(11), November 2002.

4. Foster, Y. Zhao, I. Raicu,S. Lu, Cloud Computing and Grid Computing 360-Degree Compared," **Grid Computing Environments Workshop**, 2008. GCE '08 , vol., no., pp.1-10, 12-16 Nov.2008
5. Bart Jacob, Michael Brown, Kentaro Fukui, NiharTrived, "Introduction to Grid Computing", **http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf**, First Edition, Dec. 2005.
6. Manjula K.A., Karthikeyan P., "Grid Computing-A Tool for Enhancing the Computing Power", **Indian Journal of Computer Science and Engineering**, Vol. 1, No. 1, 1-7, ISSN : 0976-5166.
7. R. Buyya, K. Bubendorfer. "Market Oriented Grid and Utility Computing", **Wiley Press**, New York, USA, 2008
8. Ahmar Abbas, Grid Computing: Practical Guide To Grid Technology & Applications Charles River | ISBN: 1584502762 | 2003 | CHM | 406 page.
9. Joshy Joseph, Craig Fellenstein, "Grid Computing", **IBM Press**, ISBN-10: 0-13-145660-1, 2004.
10. Korpela, E., Werthimer, D., Anderson, D., Cobb, J., and Lebofsky, M. SETI@home: Massively distributed computing for SETI. Comput. Sci. Engin. 3, 1 (Jan.–Feb. 2001), 79.