ISSN:0975-9646

Application of Grid Computing In Video Rendering

Dr.C.Jeyabharathi

Arulmigu Palaniandavar Arts College for Women, Palani

***Abstract*— In most organizations, there are large amounts of underutilized computing resources. Most desktop machines are used less than 5 percent of the time. In some organizations, even the server machines can often be relatively idle. Grid computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage. The processing resources are not the only ones that may be underutilized. Often, machines may have enormous unused disk drive capacity. Grid computing, more specifically, a "data Grid", can be used to aggregate this unused storage into a much larger virtual data store, possibly configured to achieve improved performance and reliability over that of any single machine. In this paper features of Grid Computing technology, and use video conversion and rendering applications to demonstrate this technology's effectiveness and high performance are discussed.**

***Keywords*— Video rendering, Grid Computing, Resource Discovery, Peer-to-Peer**

I. INTRODUCTION

The amount of videos and movies being produced is increasing rapidly. Rendering video images are complex and enhancing raw images into enhanced images is time consuming. Thus much more time is required to generate enhanced images today. If there is some way in which multiple computers could be used to generate the good quality images in parallel the time required would be reduced significantly. Grid computing paradigm is the correct choice to connect multiple computers through which required resources are available. A grid is basically a cluster of individual computers, geographically distributed, connected together to perform a few tasks in parallel. The best candidates for grid computing are applications which can be divided into tasks that can be computed in parallel, independent of any other task's result. Video rendering is one such suitable application.

Video Rendering is a time consuming task when using a single machine. The same task can be completed in much lesser time by using multiple computers. Here, the concept of a cluster of interconnected computers among which the rendering jobs are distributed. However, one of the major issues of a cluster is scalability. Keeping thousands of computers at the same location is not feasible. Going a step further, a grid of interconnected computers is considered. This grid is basically an infrastructure with interconnected nodes which are geographically distributed. This framework consists of a portal which contains the ftp server along with the load sharing facility. To the portal are connected individual nodes which contain their own sub nodes. Each node along with its sub nodes appear as one virtual node to the portal. The client uploads the frames to the ftp server. The load sharing facility assigns video rendering jobs to the individual nodes connected to the portal. Once these frames are rendered, they are uploaded back to the FTP server from where the client can download the rendered images. Grid-based computer animation rendering and scheduling divisible loads in the dynamic heterogeneous grid environment are the important tasks in multimedia environment.

II. GRID RESOURCE DISCOVERY

Resources on grid systems are widely distributed and heterogeneous in comparison to traditional and cluster systems. Resource discovery is a key grid management tool for extraction of resource information in a grid environment. Among various Resource Discovery techniques, Centralized systems, Hierarchical system and Peer-to-Peer systems are familiar in Grid Resource Discovery.

The grid resource discovery using *centralized systems* [1] provide grid middleware developers an easy to use interface to manage grid resources. They keep grid resource information by using centralized databases. In a large scale grid environment, the centralization of the service may easily create bottlenecks on the central servers. The bottleneck problem may arise both because of frequent resource updates or a large number of query requests waiting to be processed. The centralization causes another important problem in dynamic grids as being a single point of failure. In some approaches, the idea of replication of central servers is depicted in order to eliminate single point of failure. But replication of servers in a large scale dynamic grid may be very expensive in terms of communication costs. Most of these systems do not support dynamic-attribute queries because the update of dynamic resource attributes are held in discrete intervals.

The *hierarchical systems* [1] based grid resource discovery algorithms provide a more scalable platform than the centralized ones and still provide a simple user interface to manage grid resources. In a large scale grid environment, the hierarchical topology of the service decreases the probability of bottleneck problem. But single point of failure problem still exists since failure of one of the master servers in the system may cause a large part of the nodes becoming invisible to the queries.

The *Peer-to-Peer* paradigm [2] is based on the principle that every component of the system has the same responsibilities acting simultaneously as a client and a server, as opposed to the traditional client-server model. P2P systems are divided into two main categories based on the connection protocol they employ and the way, peers are organized *structured* and *unstructured*. Several hybrid approaches have been proposed to overcome the drawbacks of the two main approaches while retaining their benefits.

In *unstructured systems* [2, 3] each peer is randomly connected to a fixed number of other peers and there is no information about the location of files. There is neither a centralized directory nor any control over the network topology or resource placement. When a new peer joins the P2P network, it forms connections with other peers freely. Generally, unstructured overlays have loose guarantees for resource discovery, and it is possible that a file is not found though it exists in the network.

Structured P2P systems [2] employ a rigid structure to interconnect the peers and to organize the file indices. Structured P2P systems are equipped with a distributed indexing service which is based on hashing, and is known as Distributed Hash Table (DHT). Peers and files are mapped, usually through the same hash function, to a key space. Most structured P2P systems support naturally exact match queries in O(logN) hops, where N is the size of the key space, and range queries. However they do not support directly keyword searches which constitute the core of queries in real P2P systems. *Chord, Can, Pastry and Tapestry* are DHT based protocols which are the hot research topics of the current era.

In this paper an application for video rendering is created in VC++ and tested in online. The nodes are manually seleted for testing since there are a limited number of nodes in our Computer Labs. But when it is implemented in real Grid environment, the P2P resource discovery algorithms such as FZ-Chord [4] and Geo-Chord [5] will efficiently find the avilable node and perform video rendering easily.

III. Application Of Grid Computing In Video Rendering

The recent popular computing technique called grid computing is used in video conversion and negative images into positive image conversion. This paper demonstrates effectiveness of this technology and its high performance.

In this paper, an application for video rendering is developed and tested in real time environment. In this software all the different grid members are arranged according to a specific topology. Two categories of nodes are employed in this application namely Head node and Child node. The node which initiates the work and which needs the resources from other nodes are treated as Head node. Child node supplies resources and both Head and Children nodes which are connected in Grid together complete the given job. The node power of every node will depend on the number of direct connections it has. The client will submit a job to the root node. According to the number of child connections it has, the job will be splited equally among each of these child nodes. Now each of these child nodes will themselves split the job they have received among their children.

To test the project in real-time environment, the following platform is used.

- OS: Windows XP SP3
- IDE: Visual Studio 6.0
- Programming Language: Visual C++ 6.0
- Runtime Library: Advanced C 8.0
- Grid SDK: GIPC (Grid Image Processing Class)

GIPC uses state-of-the art technology. The GIPC class uses Windows Server to centralize actual grid. The application can be run on any computer that is connected with Internet, GIPC automatically detects the server and connects it with the node. It uses combination of HTTP, FTP & SFTP protocols dynamically and there is no need to configure the IP Address for each node. The GIPC has the facility to use mobile ad-hoc network that enables the application to deal with dynamic IP addresses, so the number of systems can be connected all over the world.

Harnessing distributed computing resources to create a so-called render farm has therefore been a solution for video makers to handle their time consuming rendering tasks. A render farm is a cluster of interconnected computers which are used for rendering computer generated imagery. There are two types of rendering methods: network rendering and distributed (split-frame) rendering. In network rendering, the images can be rendered in parallel, as each frame can be calculated independently of the others. In that case, the main communication between processors is used for uploading the initial models and textures and downloading the finished images. In distributed (split-frame) rendering, each frame is divided into tiles which are rendered in parallel.

The rendering time can be further reduced by having more computers in a cluster. However, keeping hundreds of computers in the same location requires a significant physical space. This limitation can be removed by linking up clusters of computers across different areas. This solution has brought up a new paradigm of computing called Grid computing. Computational Grid (or just Grid) can be defined as an infrastructure with interconnected nodes of resources. A node is an access point of communication or computation. The resources could be databases, application servers, networks and storage devices. The function of a Grid is to enable sharing of geographically distributed heterogeneous resources between interconnected nodes. Grid has helped many researchers and institutions in their visualization tasks such as earthquake simulation [6], visualizing speed of pollution for determining evacuation strategies [7], molecules modeling [8], volumetric image processing [9] and real time visualization of multi-physics simulation [10].

IV. Real Time Implementation

Video rendering application software is developed in VC++ and tested in real time. For testing four different nodes connected in internet are selected. Among four systems, one is taken as Grid Head and the remaining three nodes belong to Grid node type category. Two different types of processes can be done in the following manner.

1. Conversion of Raw Image into Enhanced Image
2. Conversion of Negative Image into Positive Image

Fig. 1 shows the mode selection window which contains two options namely Grid Head and Grid Node. One of the options may be selected according to our requirement.

Fig. 1 Grid Mode Selection

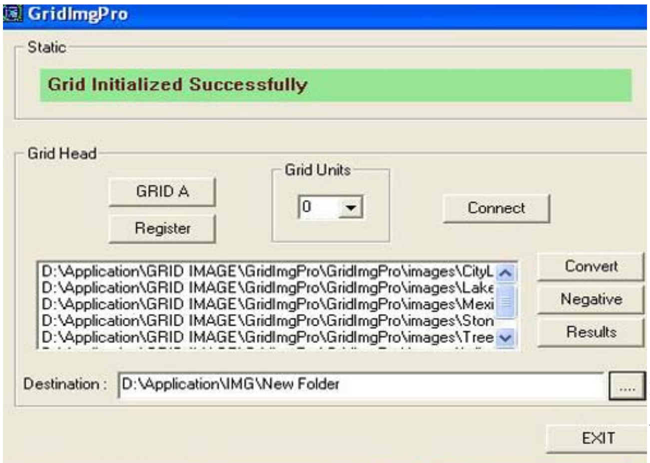

Fig. 2 Grid Head Window

The process of Grid Head selection and its corresponding execution environment is given in detail in Fig. 2. In this window, Grid Head is registered after getting proper authentication. After that Grid units in which we want to run our job should be selected. Then the connect button may be selected to have connection with the internet server. After a few seconds, Grid Head will be connected to network and the job can be completed easily.

Next to Grid Head selection process is Grid node selection process which is given in Fig.3 Here a number of Grid units must be connected to the internet through its nodeID. This number depends on the number of Grid units selected in Grid Head selection process and the number of Grid nodes available. Finally, after connecting all required grid nodes to the network the conversion process can be started based on the resource availability in Grid nodes. The job is executed in parallel (ie) images are enhanced simultaneously in all Grid units and the processing time and enhanced images are viewed as output.

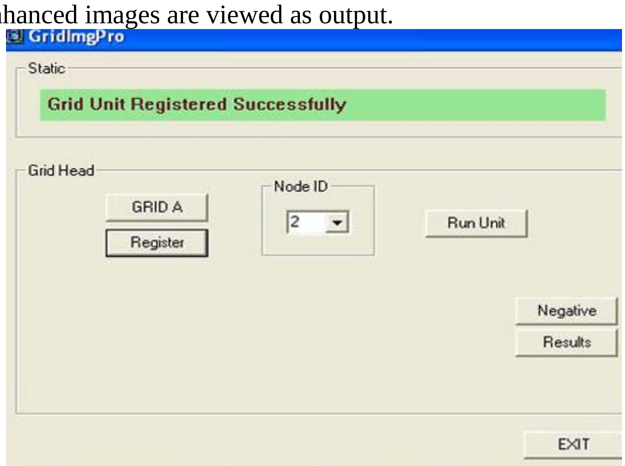

Fig. 3 Grid Node Window

The first process is executed in a single Grid node (ie, Grid Head) and its job completion time is noted. And the same job is executed in two Grid units (1 Grid Head + 1 Grid node), three Grid units (1 Grid Head + 2 Grid nodes) and four Grid units (1 Grid Head + 3 Grid nodes). The results are depicted in Table 1.

TABLE 1
RENDERING TIME FOR RAW IMAGE INTO ENHANCED IMAGE CONVERSION

No. of Raw Images	**Grid Units**	**Rendering Time (ms)**
50	4	6741
	3	8338
	2	13074
	1	25580
100	4	11406
	3	15216
	2	22804
	1	45510

Similarly the second job is also executed in a single Grid node (ie, Grid Head) and its job completion time is noted. And the same job is executed in two Grid units (1 Grid Head + 1 Grid node), three Grid units (1 Grid Head + 2 Grid nodes) and four Grid units (1 Grid Head + 3 Grid nodes). The results are depicted in the Table 2.

TABLE 2
RENDERING TIME FOR NEGATIVE IMAGE INTO POSITIVE IMAGE CONVERSION

No. of Negative Images	**Grid Units**	**Rendering Time (ms)**
50	4	6712
	3	8284
	2	12979
	1	25528
100	4	11375
	3	15166
	2	22750
	1	45500

Here, the sample input and output images are given for easy understanding. Figures shown (Fig. 4 to Fig. 9) in left side column are input images and right side column are enhanced images.

Fig. 4 Raw Image1 and Enhanced Image1

Fig. 5 Raw Image2 and Enhanced Image2

Fig. 6 Raw Image3 and Enhanced Image3

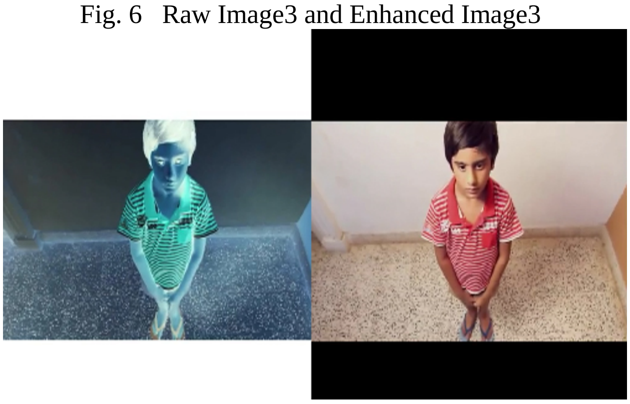

Fig. 7 Negative Image1and Positive Image1

Fig. 8 Negative Image2 and Positive Image2

Fig. 9 Negative image3 and Positive image3

The following list explains the details of the Internet Server hired in the application and the features it provide for implementing the above mentioned conversions.

- IBM/Super/HP DL 160 G8 Series
- 1 X Intel Hexa Core Xeon Processor E5-2620
- 15 M Cache, 2.0 GHz, 7.2 GT/s Intel QPI), 2x Gigabit Ethernet Card
- 6 Cores, 12 Threads
- Integrated SAS RAID 0, 1, 5
- 8 GB Fully Buffered DDR3 ECC Memory
- 2 X 1 TB SATA, 7200 RPM HDD
- 2000 GB Monthly Data Transfer
- FREE Set Up!! , 5 IP Addresses
- 24x7 Technical Support on Phone/Email/Chat
- Free Remote Reboot, Complete Remote Access
- Tier 3 Data Center in Delhi NCR

V. Conclusion

Grid computing plays a vital role in almost all areas of recent applications. In this paper, a new application is developed using Grid computing for video rendering. Image enhancement process and negative image into positive image conversion processes are tested in this application. Real-time execution of this software in network environment and corresponding results are given in detail.

References

[1] Deniz Cokuslu, Kayhan Erciyes, Abdelkader Hameurlain, "Grid Resource Discovery Based on Centralized and Hierarchical Architectures", International Journal for Infonomics (IJI), Volume 3, Issue 1, March 2010 Copyright © 2010, Infonomics Society 227.

[2] P. Trunfio, D.Talia, C. Papadakis , P. Fragopoulou, M. Mordacchini , M. Pennanen, K. Popov, V.Vlassov, S. Haridi , "Peer-to-Peer Resource Discovery in Grids: Models and Systems", This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265). Preprint submitted to Elsevier Science ,3 August 2006.

[3] Stefan Schmid and Roger Wattenhofer, "Structuring Unstructured Peer-to-Peer", Networks Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland.

[4] A.Pethalakshmi and C.Jeyabharathi, "Parallel Search in Structured Chord Protocol for Quick Resource Discovery in

Grid Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013.

[5] A.Pethalakshmi and C.Jeyabharathi, "GEO-Chord: Geographical Location Based Chord Protocol in Grid Computing", International Journal of Computer Applications, Volume 94 – No 3, May 2014.

[6] Aeschlimann, M., Dinda, P., Kallivokas, L., LOpez, J., Lowekamp, B. and O'hallaron, D. , "Preliminary Report On The Design of A Framework For Distributed Visualization", In Proceedings of The International Conference on Parallel and Distributed Processing Techniques and Applications, 1833-1839, 1999.

[7] Brodlie, K., Duce, D., Gallop, J., Sagar, M., Walton, J. and Wood, J., "Visualization in Grid Computing Environments", In proceedings of IEEE Visualization, 155-162, 2004.

[8] Bilbao-Castro, J. R., Marabini, R., Carazo, J. M., Garcia, I. and Fernandez. J. J., "The Potential of Grid Computing in Three-Dimensional Electron Microscopy", Parallel Processing Letters, World Scientific Publishing Company, 14, 2, 151-162, 2004.

[9] Muraki, S., Lum, Eric. B., Ma, K. L., Ogata, M. and Liu, X. Z., "A PC cluster system for simultaneous interactive volumetric modeling and visualization", In Proceedings of IEEE Symposium on Parallel and Large-Data Visualization and Graphics, 95-102, 2003.

[10] Suzuki, Y., Kazunori, S., Matsumoto, N. and Hazama, O, "Visualization Systems on the Information-Technology-Based Laboratory", IEEE Computer Graphics and Applications, 23, 2, 32-39, 2003.