IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 5, Ser. I (Sep. –Oct. 2021), PP 01-06
www.iosrjournals.org

A New Approach to Grid Computing for Video and Image Processing Using RMI

Md. Iqbal Aziz Khan
(Department of Computer Science and Engineering, University of Rajshahi, Bangladesh)

Abstract: *In this paper we have presented a new approach to grid computing for processing video and image using Remote Method Invocation (RMI). Our proposed grid consists of four components: Donor, Server, Client and Service. Donor allows server to use its resources so that server can provide services to the clients using the donors' resources. Donor registers itself to the server as a resource holder. Server receives request along with dataset from a client for a specific service. Server splits the dataset into smaller units and invites the registered donors for processing the smaller units. Each donor sends processed data to the server. Server compiles the processed data received from different donors and performs some basic operations on the processed data and replies the client with the result. Performance of the grid is evaluated in terms of time complexity*
Key Word*: Grid computing, RMI, Callback RMI, RMI registry, object registry.*

Date of Submission: 27-08-2021 Date of Acceptance: 11-09-2021

I. Introduction

Grid Computing is an emerging technology that provides seamless access to computing power and data storage capacity distributed over the globe. Grid computing is the application of several computers to a single problem at the same time; usually, to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. Grid computing depends on a server to divide and distribute pieces of a program or module among several computers, sometimes up to many thousands. Grid computing can also be thought of as distributed and large-scale cluster computing, as well as a form of network-distributed parallel processing [1] [2].

In the present work, grid computing is developed with RMI [3]. A lot of low configuration computers can contribute to the grid computing and make a high speed computing power. Hence working load of a server can be reduced magically using some low cost devices. Applications required high speed computing power can be carried out in the grid smoothly. On the other hand, time complexity of grid computing is inversely proportional to the number of donors. All the communications in the grid are carried out through the use of message serialization for safety and security. In this paper Section II shows the architecture of the proposed grid. Section III describes RMI and Callback RMI used in the grid. Section IV presents the process of donor registration and cancellation. Section V covers the process of submission of client request for services. Description of job scheduling is given in Section VI. Section VII covers description of the remote services provided by the proposed grid. Performance evaluation of the present work is presented in Section VIII and concluding remarks are given in Section IX.

II. Architecture of the Proposed Grid

In grid computing, at least four components are required; Server, Donor, Client and Services. Due considerations should be given to the significance of these components in implementing grid computing.

Server: This is the key component of grid computing that receives request from clients for providing the available services. Client provides dataset to the server along with the request depending on the requested service. In this case, a video or an image is sent to the server. Client can also send the dataset to the service provider or donor by avoiding the server. However, request must be submitted to the server. After receiving the dataset from a client, server splits the dataset into different units and each unit is sent to a single donor. Then donor performs operation on the received units and sends the result back to the server. After receiving the results from all the donors, server combines the results and combined result is sent to the client as a finished product. The donor containing the module for processing dataset is considered as a remote module with respect to the server as server can invoke the module. Similarly the server containing the module is considered as a remote module with respect to the client.

Donor: This component allows client to use its resources through server. Donor registers itself to the server willingly for sharing its resources in grid computing. Donor itself can also request server for a specific

service as a client. When server receives request for a specific service from a client along with the dataset, server splits the dataset into different parts according to the number of available registered donors. Server can send data to either all the registered donors or some of the registered donors. Time for processing dataset and volume of dataset is considered to decide how many registered donors should be selected. Server sends different parts of the dataset to the registered donors necessary for processing the whole dataset. Donor receives the smaller unit, performs the requested operation on the unit and sends the result back to the server. It is also possible for a single donor to provide multiple services. However, in this system only two services are considered.

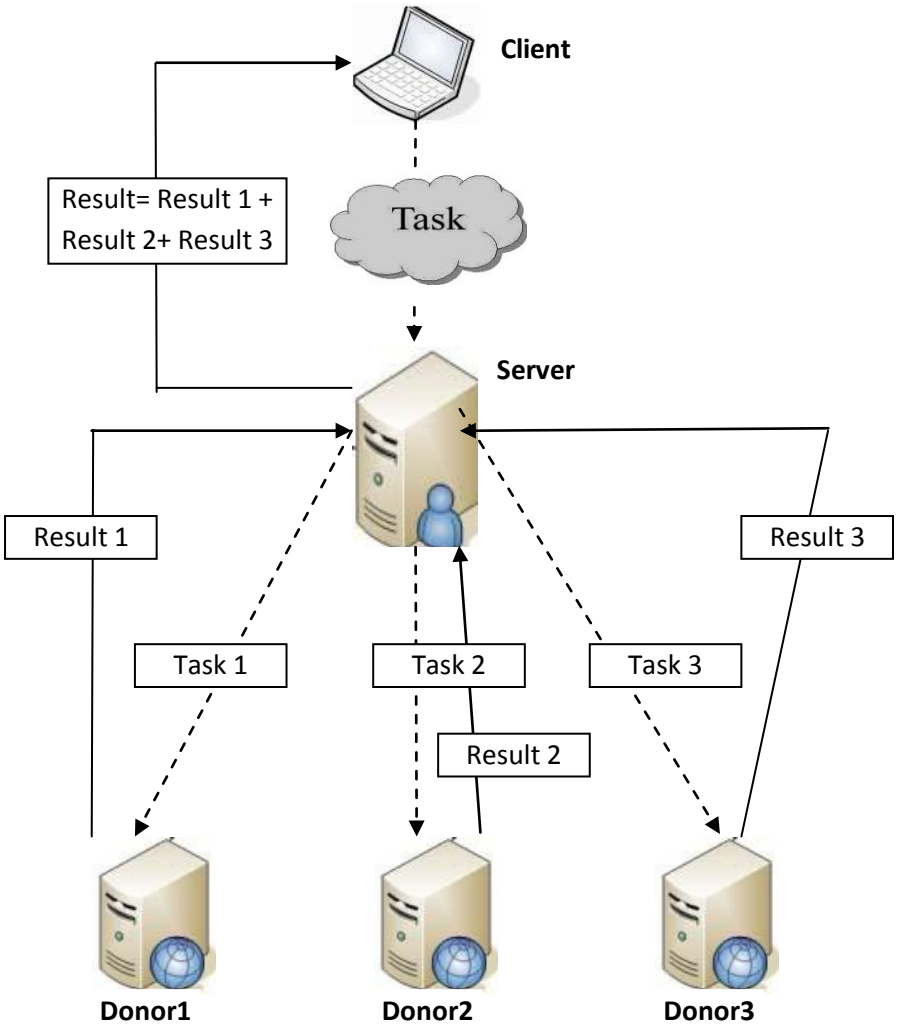

Figure-1: Architecture of the proposed Grid Computing

Client: This component of grid computing requests the server for the available services. It also sends dataset along with the request. Client receives the result from the server. Client does not know anything about the donors as server randomly selects the available donors. Client can also register itself to the server as a donor for sharing its resources in the grid computing.

Services: Server provides two services to the client, image processing and video processing. For Image Processing, an image is sent as input to the server. Server splits the image into different segments and sends the segments to the donors. Donors perform operations on the segments such as low pass filtering, high pass filtering, motion blur filtering etc. Then they send the filtered segments back to the server. Server compiles the segments received from different donors and replies the client by sending the processed image.

On the other hand for processing video, a video is sent to the server or to the donor on the basis of its size. If client sends the video to the server, server splits the video into a number of smaller units. Each unit contains a small number of frames. If client sends the video to the donor, client must split the video as per server instructions and send the smaller unit to the donors.

Service does not matter in the context of this system. Any service can be given from this system provided that donors are capable enough for the execution of the service.

III. Communication Using RMI and Callback RMI

In the proposed system, RMI is used for communication between client and server and between server and donor. Server creates a class and defines some methods in the class. Using this class it creates an object and exports it to a directory service or object registry called RMI registry [4][5]. This object is remote with respect to the client and donor. Client finds the necessary remote object in the object registry and using the reference of the remote object it can invoke the remote method in server side.

Callback RMI is used for communication between donor and server [3][6]. Server exports its remote object, IP address and port number to the object registry [7]. Each object registry maintains a list of exported objects and supports an interface for looking up these objects. Donor finds the necessary remote object in the object registry and using the reference it can invoke the remote method in server side to register itself in the server. Server only communicates with the registered donors for assigning the jobs requested from a client.

IV. Registration Process of a Donor

In the server side, two remote methods need to be provided. By invoking a remote method donor can register itself to the server. Another remote method is invoked to cancel the registration. A donor invokes remote method in server side for registration by sending its interface reference. It is necessary for the server to maintain a data structure for the interface references of donors. In this proposed grid, vector is used to maintain the references, but any suitable data structure can be used [8][9]. In the donor side, two remote methods need to be provided for two services: video processing and image processing. Donor creates an interface to declare these two remote methods. Donor sends an interface reference to the server during the registration procedure. Server inserts this interface reference into the vector. Using this interface reference server can invoke any remote method in the donor side for having a specific service [10]. Execution of this remote method is carried out using the resources of the donor [11]. When donor invokes a remote method in server side to cancel its registration, server removes the donor's interface reference from the vector.

V. Submission of Client Request for Services

Client can request server for any of the two services. For that, at first client finds the reference of the service in the RMI registry, then using the reference client can invoke a method in server for submitting a request [12]. Server contains this method for receiving the client requests and for making a queue of the client requests. Server selects the client request from the queue and forwards the request to the donor by invoking a remote method in donor side. In this system two services are available: video processing and image processing. Client requests for these services with a proper dataset. For processing video, client invokes a remote method with a video and the number of frames in the video. For having image processing service, client invokes a remote method with an image as a parameter.

VI. Job Scheduling

In the server side the received dataset is divided into different parts depending on the number of available registered donors in the grid. Job scheduling is carried our as per the following rules.

- Server divides the received dataset into different smaller units.
- Server assigns different tasks to the donor.
- Donor performs its job and sends the result back to the Client.
- If any donor cancels its registration, then server distributes the task among the remaining registered donors.
- If there is no registered donor, then server cannot provide services to the client.

VII. Remote Services

In the proposed system server provides two services to the client; a service for receiving a request for image processing and another for video processing. All services are defined as remote methods. To have a specific service client invokes remote method located in the server. In server side, this remote method receives the dataset and splits the dataset into different parts according to the number of available registered donors, and then the remote method located in the donor side is invoked with the smaller part of the large dataset. After the execution of the methods, all donors send the results of the operation performed on the individual smaller units to the server.

Client can request for the video processing service. In that case client invokes remote method in the server side by sending a video and the number of frames in the video. In the server side, the video is divided into different units according to the available registered donors, and then the remote method in the donor side is invoked with an individual part of the video and the number of frames in an individual part. In the donor side, this remote method performs the video processing requested by the server and returns the result back to the remote method of the server side. In this way server receives different results from different donors, finally it

performs some basic operations on different results received from all the donors, and returns the final result back to the client.

Similarly for image processing service, the remote method in the server side is invoked from the client side by sending an image and resolution of the image. This image is divided into different parts according to the number of interested registered donors, and then the remote method in the donor side is invoked with the small part of the image and its resolution. In the donor side, image processing operations (high pass filtering, low pass filtering etc.) are performed on the input image and the remote method in the donor side returns the result back to the remote method in the server. Server merges the processed images received from different donors to produce the same size of the original image and performs some basic operations on the whole image and sends the whole image to the client.

VIII. Performance Evaluation

As proposed grid can provide two services, a video with 1000 frames and an image with 1024x1024 sizes are used to evaluate the performance of the grid.

A video is sent from client to the server for having video processing service. Server divides the video into appropriate number of parts and passes different parts to different donors. Donors perform video processing operations on the given number of frames and send the results back to the server. Server combines the frames received from different donors in the original order and sends the final result back to the client. Figure 2 shows the service for video processing. Let us consider that a client sends a video with 1000 frames to the server and there are four registered donors in the grid. Now server splits the video into four parts i.e. First part consists of 1 to 250 frames, Second part consists of 251 to 500 frames and so on. A single part is sent to a single donor. Now let us consider that Donor3 registration is cancelled. Then server sends the video to only three registered donors Donor1, Donor2 and Donor4, and not to Donor3.

On the other hand, client can invoke image processing service by sending an image to the server. Server divides the image into smaller parts and sends different parts to different donors. Donors perform the requested operation on the smaller parts. In the proposed system motion blur filtering is applied on the smaller parts. Then the filtered image is sent back to the server. After receiving all parts of the image from different donors server merges different parts of the image and sends the processed and merged image to the client. Figure 3 shows the image processing service.

In case of video processing service and four donors time complexity can be reduced to four times. Similarly, in case of image processing service and two donors time complexity can be reduced to two times. Hence, it is realized that there is a correspondence between the number of donors and time complexity.

IX. Conclusion

In this paper a new approach to grid computing using RMI is proposed and developed for sharing resources. The performance is found with a high level of satisfactory. There are still scopes to optimize the grid for better performance. Some scopes for the future development are presented in the following section.

RMI is used in the present work. However in future, mobile agent could be used to implement grid computing. In RMI, machines are reserved for donors and clients. In case of mobile agent any machines could be selected for donors and clients during the execution of grid computing. In the present work, the same volume of job is assigned to all the available registered donors but in future job may be assigned according to the capacity of the donor. On the other hand a new format of the messages could be proposed in the future research for safe communication.

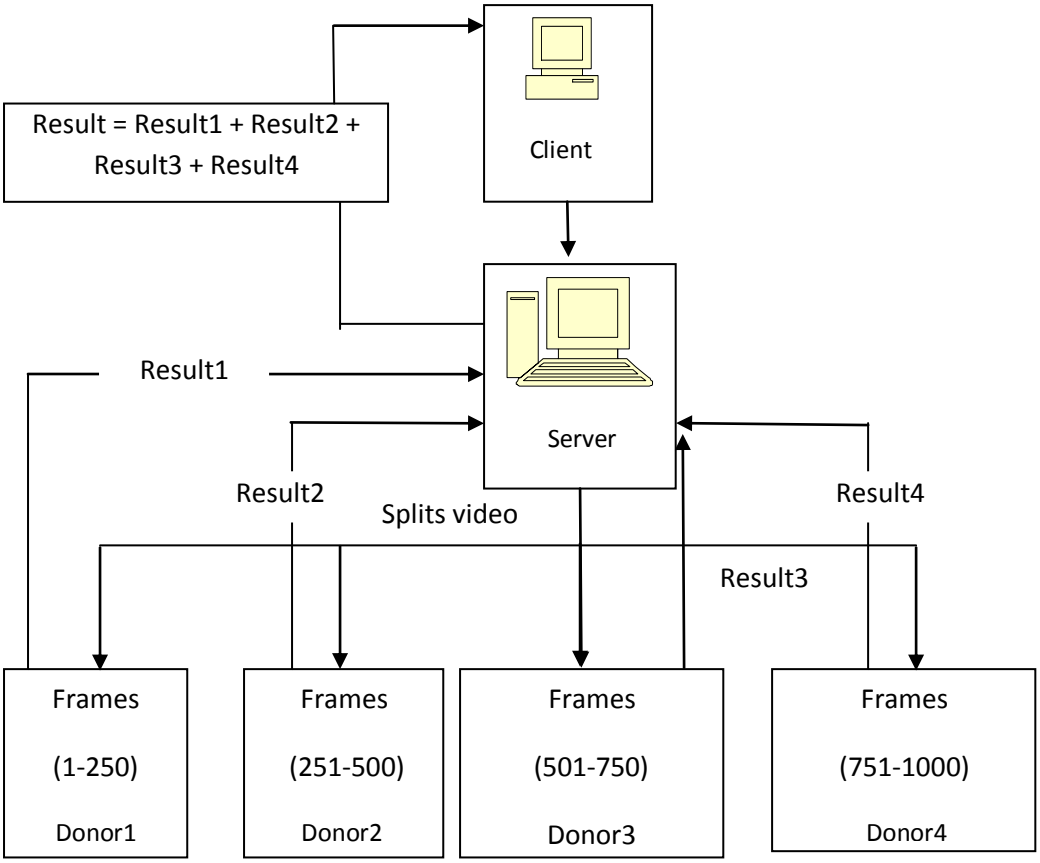

Figure 2: Video processing service

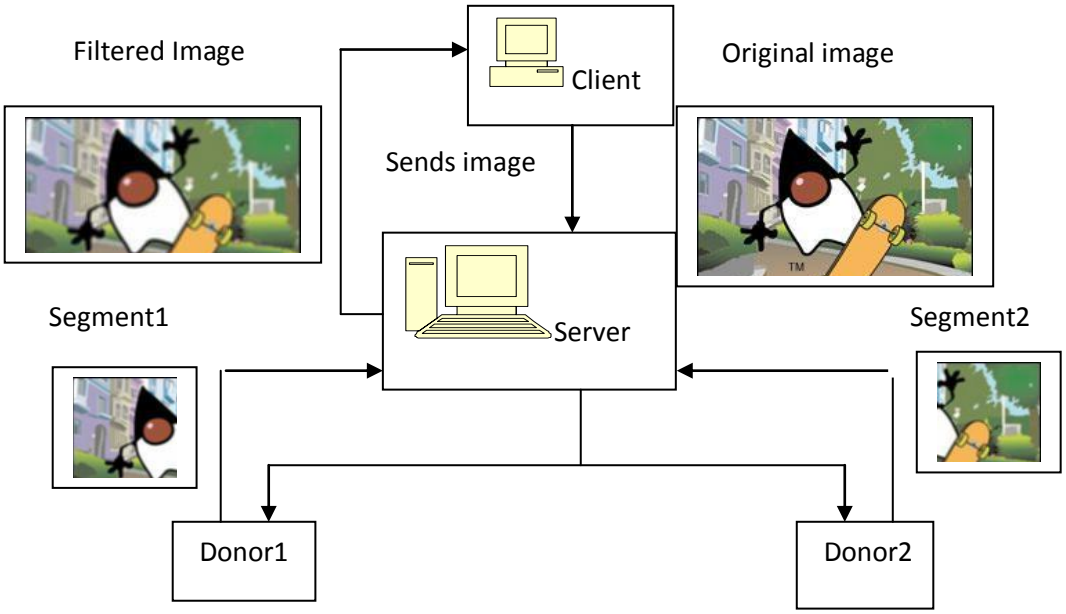

Figure 3: Image processing service

References

[1] Vladimir Silva, "Grid Computing for Developers", Charles River Media,Inc. ISBN 1-58450-424-2
[2] I. Foster, **"**What is the Grid? A Three Point Checklist", GRIDToday, July 20, 2002.
[3] M.L.Liu, "Distributed Computing PRINCIPLES and APPLICATIONS",
[4] Rajkumar Kettimuthu, William Allcock, Lee Liming, John-Paul Navarro and Ian Foster,. "GridCopy: Moving Data Fast on the Grid", Proceedings of the Fourth High Performance Grid Computing Workshop to be held in conjunction with International Parallel and Distributed Processing Symposium (IPDPS 2007), March, 2007
[5] I. Foster, C. Kesselman, J. Nick, S. Tuecke, "Grid Services for Distributed System Integration", . *Computer*, 35(6), 2002

[6] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets", Journal of Network and Computer Applications, 23:187-200, 2001 (based on conference publication from Proceedings of NetStore Conference 1999)
[7] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke," Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.
[8] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman,"Grid Information Services for Distributed Resource Sharing",Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
[9] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke., "A security architecture for computational grids.," In ACM Conference on Computers and Security, pages 83–91. ACM Press,1998.
[10] M. Livny.,"Matchmaking: Distributed resource management for high throughput computing," In Proc. 7th IEEE Symp. on High Performance Distributed Computing, 1998.
[11] I. Foster, et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration.,"Open Grid Service Infrastructure WG, GGF 2002.
[12] V. Raman, et al., "Data Access and Management Services on Grid," GGF, DAIS group 2002.

Md. Iqbal Aziz Khan. "A New Approach to Grid Computing for Video and Image Processing Using RMI." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 23(5), 2021, pp. 01-06.