ISSN No. 0976-5697

Volume 4, No. 8, May-June 2013

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A survey of Image Processing and Analysis in Distributed Environment

Pavithra K, Praveen N, Manjula K, Archana Yashodhar

Dept. of Computer Science, Mangalore University, Mangalore, Karnataka, India

vbpavithra@yahoo.in

***Abstract*:** Extensive research and development has taken place over last few years in the area of Image Processing in distributed environment. Areas to which this discipline has been applied include business, communication, agriculture, medicine, defense intelligence etc. In this paper, we provide a brief survey of recent developments in the field of Distributed Image processing. This paper also discusses some of the current Image processing technologies available in certain distributed environments

I. INTRODUCTION

In this paper, we discuss contributions of various computer professionals for distributed and parallel image processing in some specific domains.

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. The area of image processing and analysis is very complex, but of raised importance in many domains [11].

Image processing applications requirements can be best met by using the distributed environment. Image processing applications require various types of resources like high memory availability, high-resolution graphic displays, high processing power, drives for magnetic tape/cartridge/floppies, optical disk, database, etc. The resource requirement changes with time due to the availability of new and better resources. Hence, it is not possible to assume the availability of all resources on a single system. The distributed environment not only helps in optimum utilization of such resources but also helps in achieving better throughput using multiple processors in parallel [1].

Development of an application having built-in automated data transfer, capability of using multiple machines in parallel and robust error handling is a challenging job. Various technologies like Java Remote Method Invocation (RMI), Common Object Request Broker Architecture (CORBA), Distributed Component (DCOM), Remote Procedure Calls (RPC), Simple Object Access Protocol (SOAP), Parallel Virtual Machines (PVM), Message Passing Interface (MPI) and recently grid computing are available to enable distributed processing [2].

II. OVERVIEW OF FEW DISTRIBUTED IMAGE PROCESSING APPLICATIONS

Paper [2] presents a distributed image processing application, which has the objective to create an efficient and powerful instrument for image analysis in varied domains like medical applications, industrial applications and others. This paper implements a platform-independent distributed Image Processing System which is Open Source software and is registered by SourceForge.net, provides free hosting to Open Source software development projects. This application has Client-Server architecture, using CORBA technology and C++ and Python as programming languages.

Client-Server application is developed using CORBA and XML. CORBA is a standard for distributed computing and systems integration, which allows lightweight objects to be accessed from anywhere in a network without concern for the operating system they are running on or the programming language they are written in. XML is a simple, very flexible text format derived from the Standard Generalized Markup Language (SGML). XML is playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere for its application neutrality, platform neutrality, hierarchical structure and international language support.

The image processing application is divided in two parts. The first part is the data processing part, which requires much computing power is implemented in the server using C++ programming language. The second part is the presentation part and is implemented in the client using Python programming language. The transfer of the multimedia files becomes a critical point in this system, mainly because of the conversion of binary image data to text (ASCII) format in order to be serialized in XML files. In order to optimize the system, two encoding algorithms, base64 and hex encoding are used. When a succession of image transformations should be done for an image, instead of transferring the intermediary images from client to server and server to client for every process, the image is transferred only one time from client to server, at the beginning of the processes succession, and is received back from server after all processes are done. This also helps to optimize the system.

In [3], the authors have presented a partial evaluation technique to reduce communication costs of distributed image processing. The aim of this work was to demonstrate the performance benefits of incomplete information processing and partial evaluation can be brought to distributed image processing in a high level manner that is transparent to users. Partial evaluation is an automatic program transformation technique which allows partial execution of a program by pre-computing parts of the program that depends on known input parameter settings, producing a residual program that depends on the rest of the inputs. It enables the construction of specialized program from general highly parameterized software systems. Partial evaluation has been used in such areas as compilation, scientific computing, meta-programming, computer graphic, pattern matching, and others [3][4]. In this work, partial evaluation is used not only to remove much of the excess overhead of sequential and shared memory parallel computation, but also in distributed application to reduce the number of messages when some information about the image or part of the image is known. The work demonstrates that good parallel speedups are attainable using MPI and can be integrated into existing distributed environment. 1D-Fast Fourier and 2D Haar wavelet transforms' optimization is presented.

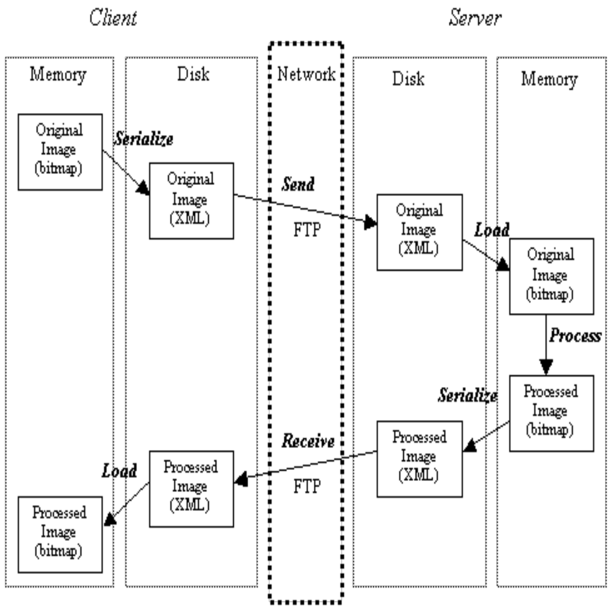

Figure 1: The information flow in Remote Processing [2]

Two main steps are considered here: pre-processing and message elimination. At the preprocessing step, the MPI code is transformed to facilitate detection of static memory accesses. The main parallelized code is replicated in accordance with the number of processes given, constants are propagated, dead codes are eliminated, and loops are unrolled. At the message elimination step, static memory accesses are evaluated by inserting a special instruction into the corresponding processes to perform the remote request locally. After evaluation of static memory requests, at the second step, partial evaluation of local codes is performed to complete execution of requests that have enough information.

The code size and memory consumption may be increased while improving efficiency. The extra code generation can be avoided for some extent by restriction of unfolding, recursion depth, and loop unrolling automatically or with human interaction during partial evaluation step. In this work [3], to increase message passing performance, the elimination of synchronization issues, nonstrict asynchronous data access, and the reduction of the number of messages are applied. Experimental results of Fast Fourier and Haar wavelet transforms presented in this paper [3] show the good speedup of partially evaluated MPI programs and demonstrate that the performance benefits of incomplete information processing can be brought to distributed image processing in a high level manner.

In addition to the description of the proposed optimization technique, this paper [3] also discusses partial evaluation in distributed data processing, incomplete data structures and incomplete data structure memory system.

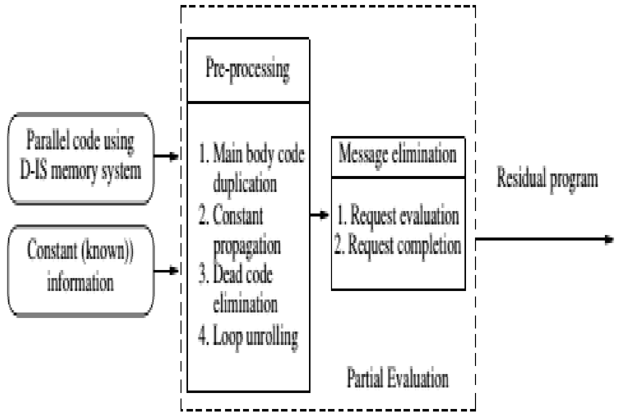

Figure 2: General view of distributed programs partial evaluation [3]

The rapid growth of distributed image processing and he number of users in this field requires systems which are easy to use. Most available image processing systems are based on processing. The user himself has to choose the proper procedure and apply it to his data. Often, several procedure blocks must be applied in a certain sequence to get the desired result. This requires in-depth knowledge of the image processing system, because it is by no means trivial to choose the proper function at a certain stage of processing. If a good way of "how to process" was found, it is difficult or even impossible to store the processing sequence within the system. Franz Niederl and Alois Goller [5], present a data driven distributed image processing system. They have discussed design considerations concerning the interaction of the back-end of a distributed image processing system with other system components, and strategies for effective job distribution. The user specifies the input data and the desired result and the system automatically searches for possibilities to execute this job, proposes several solutions and dispatches the job to the selected components. Scheduling, network performance and charging are some of the observed problems. Java RMI, CORBA and distributed queuing systems (DQS) were investigated for their qualification as scheduler and broker.

This data driven system, which is able to execute different methods on different distributed computers consists of four components: WWW-Server, Graphical User Interface, Broker and Back-end. Fig. 3, given below, shows these components of the proposed system. Since almost all programs have their own GUI installed locally, we require only a common WWW-browser. The broker is the link between GUI and the back-end, which can be realized by a combination of a specific server, for instance a Java-server, and a scheduler. The back-end is the processing part of the system. It is a collection of distributed tools like methods, single and multi processor machines, hard-disks and so on, which can work together in many variable combinations. This platform independent image processing system is much easier to handle and turns more attention to the data and the desired results than existing systems do. The broker can be set up such that it is able to construct automatically new method sequences whenever no method base entry can be found for the current problem. In addition to this, more research on scheduling and load balancing may help improving the system.

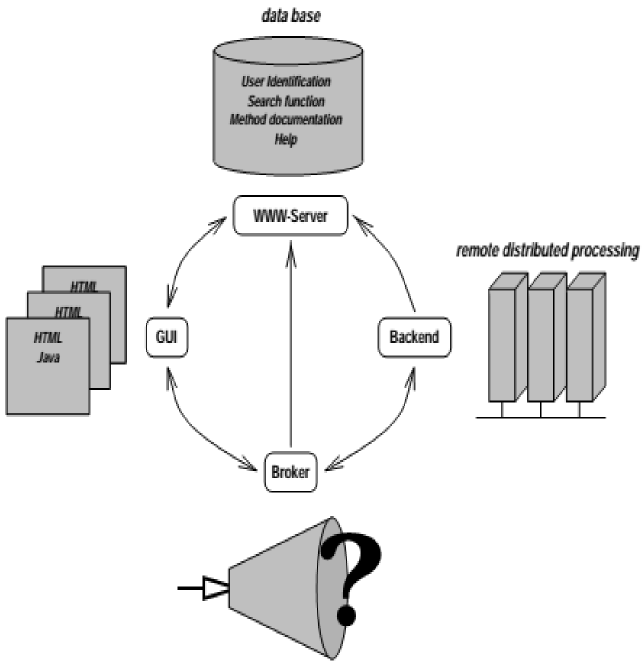

Figure 3: The four main components of the system [5]

The paper [6] provides a well organized information about Content-based Image Retrieval. The authors discuss some of the recent key contributions related to image retrieval and automated image annotation and also point out some of the key challenges involved in the adaptation of existing image retrieval techniques to build useful systems that can handle real-world data.

In distributed environment, a critical concern is to efficiently partition and to schedule the tasks among available processors in such a way that the overall processing time of the submitted tasks is at a minimum. X.L.Li, B.Veeravali, and C.C.Ko conducted experiments on a new paradigm, Divisible Load Theory (DLT) to process very large amount of data [9]. Low-level pixel-wise processing of an image for several application requirements, in general, requires a lot of CPU power. Image partitioning, and computing on a network of computers is a natural solution to minimize the processing time. As image data are divisible in nature, they naturally fit in the DLT research domain [9]. As experiment example, they considered edge detection using Sobel detectors. The experimental results showed the benefits of using the divisible load paradigm in a distributed computing network with high-speed machines and a link.

The three main goals of any distributed computing system are: (i) efficient and selective processing of data, (ii) support network collaboration without clogging distribution networks; and (iii) allow transparency of experiments through repeatability and verifiability of experiments [7]. The paper [7] discusses the architecture and implementation of a framework in Astro-WISE, an astronomical approach to distributed data processing. The authors show that lineage data collected during the processing and analysis of datasets can be reused to perform selective reprocessing at sub-image level on datasets while the remainder of the dataset is untouched. With the help of their observation, they have tried to prove that sub-image processing is a powerful methodology that will provide efficient solutions for what are otherwise manual, time-consuming tasks. However, there is a scope for further work in this field.

III. CONCLUSION

In this paper, we have tried to highlight major recent advancements and scope for future work in the area of Distributed image processing. As we know, the image processing requires much of processing power. In any distributed Image Processing application, more attention should be paid on communication operations than computational steps. In distributed environment , network latency also affects the power of execution of particular operations. Since security and reliability are major concerns in distributed environment, there is also a need for adopting some security algorithms in distributed image processing.

IV. REFERENCES

[1] Haresh S. Bhatt, V. H. Patel, A. K. Aggarwal, “Web Enabled Client-Server Model for Development Environment of Distributed Image Processing”, First IEEE/ACM International Workshop Bangalore, India, December 17, 2000 Proceedings.

[2] “DISTRIBUTED IMAGE PROCESSING APPLICATION CONSIDERING CORBA AND XML TECHNOLOGY” Retrieved from: imageprocess.sourceforge.net/corba_python_2.htm, Last Accessed on: 19/05/13.

[3] A. Tchernykh, A. Cristóbal-Salas, V. Kober, and I. A. Ovseevich, “ Partial Evaluation Technique for Distributed Image Processing”, ISSN 1054-6618, Pattern Recognition and Image Analysis, 2007, Vol. 17, No. 3, pp. 390–398.

[4] C. Consel and O. Danvy, “Partial Evaluation of PatternMatching in String,” Information Processing Letter 30(2), 79–86 (1989).

[5] Franz Niederl, Alois Goller, “Method Execution on a Distributed Image Processing Back-end”, Parallel and Distributed Processing, 1998. PDP '98. Proceedings of the Sixth Euromicro Workshop on 23 Jan 1998

[6] Ritendra Datta, Jia Li, and James Z. Wang, “Content-Based Image Retrieval - Approaches and Trends of the New Age”, MIR’05,November 11-12, Singapore, 2005.

[7] Johnson Mwebaze, John McFarland, Danny Booxhorn, “A Data Lineage Model for Distributed Sub-image Processing”, SAICSIT ’10, October 11-13, Bela Bela, South Africa Copyright 2010.

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang, “Image Retrieval: Ideas, Influences, and Trends of the New Age”, ACM Computing Surveys, Vol. 40, No. 2, Article 5, Publication date: April 2008

[9] X.L.Li,B.Veeravali, and C.C.Ko, “Distributed image processing on a network of workstation”, 2003 international journal of computers and applications, vol 25,no 2.

[10] Byeong-Ho KANG, “A Review on Image and Video processing”, International Journal of Multimedia and Ubiquitous Engineering Vol. 2, No. 2, April, 2007.

[11] KING-SUN FU, AZRIEL ROSENFELD, “Pattern Recognition and Image Processing”, IEEE TRANSACTIONS ON COMPUTERS, VOL. C-25, NO. 12, DECEMBER 1976.