Efficient Collective Communication Paradigms for Hyperspectral Imaging Algorithms Using HeteroMPI

David Valencia[1,⋆], Antonio Plaza[1], Vladimir Rychkov[2],
and Alexey Lastovetsky[2]

[1] Technology of Computers and Communications Dept.,Technical School of Cáceres
University of Extremadura, Avda. de la Universidad S/N, E-10071 Cáceres (Spain)
{davaleco,aplaza}@unex.es
[2] Heterogeneous Computing Laboratory, School of Computer Science and Informatics
University College Dublin, Belfield, Dublin 4, Ireland
{vrychkov,alastovetsky}@ucd.ie

Abstract. Most of the parallel strategies used for information extraction in remotely sensed hyperspectral imaging applications have been implemented in the form of parallel algorithms on both homogeneous and heterogeneous networks of computers. In this paper, we develop a study on efficient collective communications based on the usage of HeteroMPI for a parallel heterogeneous hyperspectral imaging algorithm which uses concepts of mathematical morphology.

Keywords: Hyperspectral Imaging Algorithms, HeteroMPI.

1 Introduction

Hyperspectral imaging identifies materials and objects in the air, land and water on the basis of the unique reflectance patterns that result from the interaction of solar energy with the molecular structure of the material[1]. Most applications of this technology require timely responses for swift decisions which depend upon high computing performance of algorithm analysis. Examples include target detection for military and defense/security deployment, urban planning and management, risk/hazard prevention and response including wild-land fire tracking, biological threat detection, monitoring of oil spills and other types of chemical contamination. These images are characterized by covering tens or even hundreds of kilometers long, having hundreds of MB in size. Few consolidated parallel techniques for analyzing this kind of data currently exist in the open literature, and mainly all of them implemented on homogeneous networks of computers using MPI. Although the standard MPI[3] has been widely used to implement parallel algorithms for Heterogeneous Networks of Computers (HNOCs), it does not provide specific means to address some additional challenges posed by these networks, including the distribution of computations and communications unevenly,

---

[⋆] Corresponding author.

A. Lastovetsky et al. (Eds.): EuroPVM/MPI 2008, LNCS 5205, pp. 326–331, 2008.

taking into account the computing power of the heterogeneous processors and the bandwidth of the communications links. To achieve these goals, HeteroMPI was developed as an extension of MPI which allows the programmer to describe the performance model of a parallel algorithm in generic fashion[4], a very useful feature for heterogeneous hyperspectral imaging applications to define distribution of workload and communications, which typically make intensive use of scatter/gather communication operations).

In this paper, our main goal is to study on several approximations for efficient collective communications adapted to the particularities of a heterogeneous hyperspectral image processing scenario already developed using HeteroMPI, basing our developments on the communication model by Lastovetsky et al.[9]. The paper is structured as follows. Section 2 first describes hyperspectral imaging algorithm considered in this study and main features of HeteroMPI. Section 3 explore the different paradigms studied. Finally, section 4 concludes with the experimental results obtained and some remarks and hints at plausible future research.

2 Related Work

Several hyperspectral imaging algorithms have been implemented using MPI as a standard development tool. Examples include the distributed spectral-screening principal component transform algorithm (S-PCT)[6], D-ISODATA[7], a computationally efficient recursive hierarchical image segmentation algorithm hybrid method (called RHSEG)[8], and a morphological approach for classification of hyperspectral images called automated morphological classification (AMC)[10], which takes into account both the spatial and the spectral information in the analysis in a combined fashion. An MPI-based parallel version of AMC has been developed and tested on NASA's Thunderhead cluster[12], showing parallel performance results superior to those achieved by other parallel hyperspectral algorithms in the literature[2]. In particular, this algorithm is the one used in our experiments because it is an exemplar algorithm with the main characteristics of the different hyperspectral imaging existing in the literature. An important limitation in the mentioned parallel techniques is that they assume that the number and location of processing nodes are known and relatively fixed, allowing the use of the standard MPI specification. This approach is feasible when the application is run on a homogeneous distributed-memory computer system. However, selection of a group for execution on HNOCs must take into account the computing power of the heterogeneous processors and the speed/bandwidth of communication links between each processor pair[5]. This feature is of particular importance in applications dominated by large data volumes such as hyperspectral image analysis, but is also quite difficult to accomplish from the viewpoint of the programmer. The main idea of HeteroMPI is to automate and optimize the selection of a group of processes that executes a heterogeneous algorithm faster than any other possible group.

Particularly, HeteroMPI has been used to measure the processing power of each processor in the moment the execution of the heterogeneous algorithm is to be made. To measure this, the directive **HeteroMPI_Recon** has been used along with a benchmark defined to reflect the most important features of the real algorithm in terms of computational cost and to stress and activate the whole memory hierarchy. Then, with directive **Hetero_Group_create** and performance model defined through mpC[11], the best heterogeneous executing group is created, and data is distributed based on the actual processing power available at each node.

3 Communication Patterns

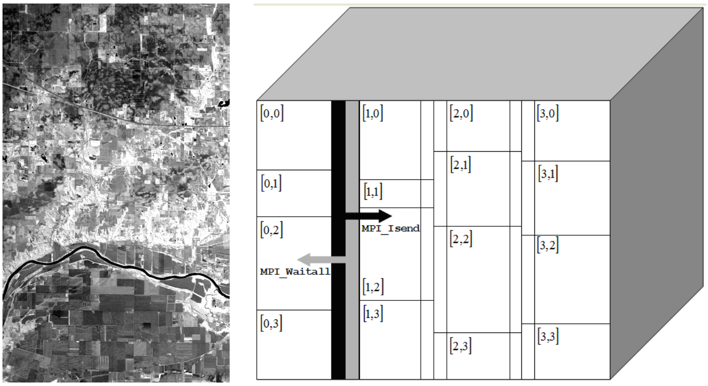
Recently, Lastovetsky et al.[5][9] designed a new model for describing performance of all collective communications that generally take place in parallel MPI applications and, in particular, in those applications executed on heterogeneous clusters based on a switched Ethernet networks. The idea is to model a few simple parameters with point-to-point communication between each pair of nodes on the network, and then use these parameters to build an estimate for collective communications based on a one-to-many and many-to-one pattern. In particular, in this paper we have further studied different solutions to the problem of sending information with different sizes located on the limits of partitions between processes (see Fig. 1(b)), whose size is located on the congestion area predicted by the communication model. The communication paradigms considered are: Chaotic Non-Blocking (CNB), Divided Chaotic Non-Blocking (DCNB) and Subgroup-Based (SB) Communications. CNB is characterized as a naive approximation, with highly balanced computing phase (thanks to the benchmark and directives of HeteroMPI) and the use of non-blocking communication directives for overlapping. DCNB is developed with the idea of coping with the problem of having communications located on the congestion region. In order to evade the congestion region predicted by the model in the network, it is necessary to introduce very complex control code to correctly retrieve the data, also making it completely independent of the particular algorithm, thus only dependent on the parameters of the network and the size of the message passed to the communication framework, posing as a robust algorithm for subdivision of messages and ordered reconstruction upon reception that evades the congestion area. On the other hand, SB is developed with the idea of evade control code and make use of divided messages. Introducing an ordered communication pattern by means of subgroups of processes and collective communications we eliminate the need of control code.

4 Experimental Results

In the present section, we describe the images and heterogeneous cluster used in our studies, along with a comparison of the communication times obtained for the different communication frameworks mentioned before.

4.1 Heterogeneous Cluster and Hyperspectral Image

The heterogeneous cluster used is located in the Heterogeneous Computing Laboratory of the University College Dublin. It is formed by 16 different machines



(a) (b)

Fig. 1. (a) Spectral band at 587 nm wavelength of an AVIRIS scene comprising agricultural and forest features at Indian Pines, Indiana. (b) Communication of a shared part of the hyperspectral image between neighboring processes.

interconnected by two level 5 Cisco switchs that allows hardware reconfiguration of bandwidth between nodes. The processors are as follows: one IBM x306 3.0GHz AMD processor; two IBM x326 2.2GHz AMD processors; two Dell PowerEdge SC1425 Xeon processors at 3.0GHz and 2.2GHz; 6 Dell PE750 Pentium 3.4GHz processors; 3 HP DL140 Xeon Processors at 2.8GHz, 3.4GHz and 3.6GHz; two HP DL320 Celeron at 2.9GHz and 3.4GHz Pentium 4 Processors. The cluster is connected via an Ethernet switch with adjustable bandwidth (from few Kilobytes) on each link. In this research, we have only used 15 machines due to a problem of disk space in node 2 during experiments.

The image used in the experiments is characterized by very high spectral resolution (224 narrow spectral bands in the range 0.4-2.5 μm) and moderate spatial resolution (614 samples, 512 lines and 20-meter pixels). It was gathered over the Indian Pines test site in Northwestern Indiana, a mixed agricultural/forested area, early in the growing season. Fig. 1(a) shows the Indian Pines AVIRIS hyperspectral data set considered in experiments. The data set represents a very challenging classification problem and it is a scene universal and extensively used as benchmark to validate classification accuracy of hyperspectral imaging algorithms.

4.2 Communication Times

Our experiments have focused on the measurements of the communication times for each paradigm used on communicating the data located on the borders of each partition assigned to the different processor on a processing power basis, producing thus different number of messages and sizes. Each execution has been made with the same group of processors, only varying the data assigned due to particular processing load at each node, except in the case of SB, where additional subgroups are created to scatter the data from the borders.

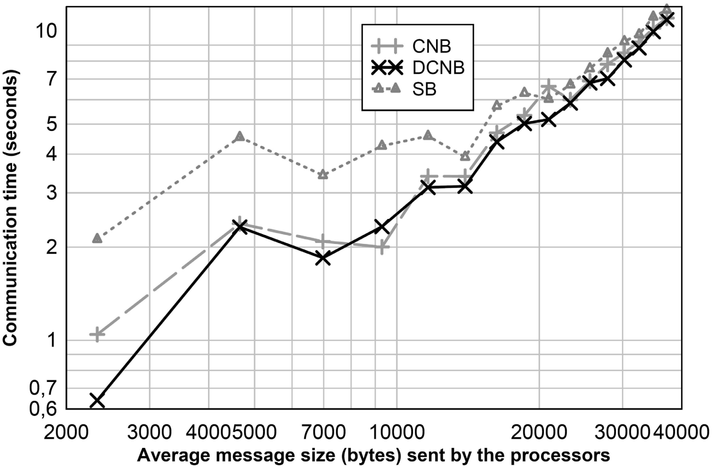


Fig. 2. Mean communication time for each particular communication paradigm

In Fig. 2, we show the mean communication time of the 15 machines using each one of the communication paradigms before mentioned. As can be seen, before reaching 6972-9296 bytes the CNB method is similar to DCNB, as expected from the model (we are still before the congestion area in most of the processors which is located around 3-4KB), thanks to small messages and no overhead for control in this implementation. Once we reach 9296 bytes, all the processes enter the congestion region, occurring then the effects of non-linearity in the communications[9]. Now, the best results are obtained by the DCNB. This is due to the use of division of the original message into several smaller messages that will fall out of the congestion area. Even though the overhead introduced with the control code, this implementation gives the best results, showing that the division of messages poses as a key solution to the problem itself. Also from the figure, we can see that the times of the SB are worst than those of the DCNB, but still very close, specially when the size of messages reach the congestion area, due to elimination of control overhead and ordered nature imposed by groups and Scatter operations. This is a very promising solution to the communication problem studied in this paper, upon the inclusion of non-blocking divided collective communications and overlapping groups.

In general, the best results are those of the DCNB, but all the paradigms show a logarithmic scaling behavior and approximation between values due to higher message sizes and overhead of the network, until the linearity is regained when reaching 65KB (as predicted by the model).

5 Conclusion

The aim of this paper has been the study of different collective communication paradigms for its use on the implementation of parallel hyperspectral imaging algorithms on heterogeneous networks of computers, using for it HeteroMPI library and communication models. The results obtained are very promising and reveal different solutions and approaches varying in complexity. As future work, we plan to integrate subgroups and collective nonblocking scatter/gather operations which may allow us to resolve the problem of excessive communications in the congestion area

References

1. Chang, C.-I.: Hyperspectral imaging: Techniques for spectral detection and classification. Kluwer, NY (2003)
2. Plaza, A., Valencia, D., Plaza, J., Martinez, P.: Commodity cluster-based parallel processing of hyperspectral imagery. Journal of Parallel and Distributed Computing 66, 345–358 (2006)
3. Dongarra, J., Huss-Lederman, S., Otto, S., Snir, M., Walker, D.: MPI: The complete reference. MIT Press, Cambridge (1996)
4. Lastovetsky, A., Reddy, R.: HeteroMPI: Towards a message-passing library for heterogeneous networks of computers. Journal of Parallel and Distributed Computing 66, 197–220 (2006)
5. Lastovetsky, A., Mkwawa, I., O'Flynn, M.: An accurate communication model for a heterogeneous cluster based on a switched-enabled Ethernet network. In: The 12th International Conference on Parallel and Distributed Systems (ICPADS) (2006)
6. Achalakul, T., Taylor, S.: A distributed spectral-screening PCT algorithm. Journal of Parallel and Distributed Computing 63, 373–384 (2003)
7. Dhodhi, M.K., Saghri, J.A., Ahmad, I., Ul-Mustafa, R.: D-ISODATA: A distributed algorithm for unsupervised classification of remotely sensed data on network of workstations. Journal of Parallel and Distributed Computing 59, 280–301 (1999)
8. Tilton, J.C.: Parallel implementation of the recursive approximation of an unsupervised hierarchical segmentation algorithm. In: Plaza, A., Chang, C.-I. (eds.) High-performance computing in remote sensing. Chapman & Hall/CRC Press, Boca Raton (2007)
9. Lastovetsky, A., O'Flynn, M., Rychkov, V.: Optimization of Collective Communications in HeteroMPI. In: Cappello, F., Herault, T., Dongarra, J. (eds.) PVM/MPI 2007. LNCS, vol. 4757, pp. 135–143. Springer, Heidelberg (2007)
10. Plaza, A., Martinez, P., Plaza, J., Perez, R.: Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. IEEE Transactions on Geoscience and Remote Sensing 43(3), 466–479 (2005)
11. Valencia, D., Lastovetsky, A., Plaza, A.: Design and implementation of a parallel heterogeneous algorithm for hyperspectral image analysis using HeteroMPI. In: Proceedings of the Fifth International Symposium on Parallel and Distributed Computing (ISPDC 2006), Timisoara, Romania (2006)
12. NASA's Thunderhead Cluster webpage, `http://thunderhead.gsfc.nasa.gov/`