IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 4, November 2012)

An Optimized Mapping Strategy for Parallel Image Processing

Vithika Goyal C. S. Lamba

vithikagoyal@yahoo.co.in *professorlamba@gmail.com*

***Abstract*— Speed has been the most concerning part of technical computing and image processing is one of the most popular research topic today. This paper presents an optimized parallel image processing over biological images using genetic algorithm. The result is compared with conventional parallel processing.**

***Keywords*— Parallel processing, Genetic Algorithm, Speed, Optimization.**

I. INTRODUCTION

Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel"). There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by computers has become a concern in recent years, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multicore processors [1]. A NUMBER of parallel computer architectures, where several processing elements (PE's) are connected by an interconnection network, have been proposed or built in response to the ever-growing need for speeding up computationally intensive tasks. Most of these architectures may be classified into two groups. One group of architectures, called dedicated architectures, aims at maximizing the achievable performance for a particular task or a class of similar tasks. Usually, there exists relatively little room for optimizing the assignment of decomposed subtasks on the dedicated architecture, i.e., scheduling. The architectures of the other type, called general-purpose architectures, are designed so that they can provide a good average performance for a broad range of tasks. Therefore, scheduling becomes an important problem for this type of architectures since it has a substantial effect on system performance and utilization [2].

Various approaches to the multiprocessor scheduling problem have been proposed Because of the intractability of the problem, heuristic approaches have been developed to solve the problem. Kashara and Narita proposed a heuristic algorithm (critical path/most immediate successors first) and an optimization/approximation algorithm (depth first/implicit heuristic search). Chen *et al.* developed a state-space search algorithm (A *) coupled with a heuristic derived from the Fernandez and Bussell bound to solve the multiprocessor scheduling problem. Hellstrom and Kanal map the multiprocessor problem into a neural network model, asymmetric mean-field network. In this paper, we present a genetic algorithm approach to the multiprocessor scheduling problem [3]. The efficiency of a parallel computing system is commonly measured by completion time, speedup, or throughput, which in turn reflect the quality of the scheduler. The scheduling problem is known to be NP-complete for the general case and even for many restricted cases. For this reason, scheduling is usually handled by heuristic methods which provide reasonable solutions for restricted instances of the problem.

This paper presents the optimization of scheduling for parallel image procesing by genetic algorithm. The genetic algorithm is discussed in section II and the proposed methodology in section III, while section IV shown the simulation and results.

II. GENETIC ALGORITHM

A Genetic Algorithm (GA) is a search algorithm which is based on the principles of evolution and natural genetics [4], [5]. It combines the exploitation of past results with the exploration of new areas of the search space. By using survival of the fittest techniques combined with a structured yet randomized information exchange, a GA can mimic some of the innovative flair of human search. A generation is a collection of artificial creatures (strings). In every new generation a set of strings is created using information from the previous ones. Occasionally a new part is tried for good measure. GAs are randomized, but they are not simple random walks. They efficiently exploit

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 4, November 2012)

historical information to speculate on new search points with expected improvement. The central theme of research on GAs has been robustness. The balance between efficiency and efficacy is necessary for survival in many different environments. The implications of robustness for artificial systems are manifold. If artificial systems can be made more robust, costly redesigns can be reduced or eliminated. If higher levels of adaptation can be achieved, existing systems can perform their functions longer and better. Features for self-repair, self-guidance, and reproduction are the rule in biological systems, whereas they barely exist in the most sophisticated artificial systems. Random search algorithms have achieved increasing popularity as researchers recognize the shortcomings of calculus-based and enumerative schemes [6], [7]. Random walks and random schemes that search and save the best must be discounted because of efficiency requirements. Random searches, in the long run, can be expected to do no better than enumerative schemes [8]. Random search methods are distinct from randomized techniques. A GA is an example of a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding of a parameter space. Simulated annealing is another example that uses a random process to guide its form of search for minimal energy states [9].

A GA starts with a pool of feasible solutions (population) and a set of biologically inspired operators defined over the population itself. At each iteration, a new population of solutions is created by breeding and mutation, with the fitter solutions being more likely to procreate. According to evolutionary theories, only the most suited elements in a population are likely to survive and generate offspring, transmitting their biological inheritance to the next generation. GAs operates through a simple cycle of stages: creation of a population a strings, evaluation of each string, selection of the best strings, and reproduction to create a new population [4], [10], [11], [12], [13]. Individuals are encoded as strings known as chromosomes composed over an alphabet. The chromosome values, genotypes, are uniquely mapped onto the decision variable, phenotypic domain. The most common representation for GAs is the binary alphabet {0; 1}. Other representations include ternary, integer, and real valued [14]. Variables are mapped onto the chromosome. When the chromosome is decoded into its phenotypic values, meaning specific to the problem can be gained. Once the chromosome has been decoded, it is possible to evaluate the performance, or fitness, of individuals in a population. An objective function is used to characterize an individual's performance to the problem. This is analogous to an individual's ability to survive in the natural world. Thus, the objective function gives the basis for selection of pairs of individuals that will be mated together during reproduction. During selection, each individual is assigned a fitness value given by the objective function. Then pairs are selected for matting. Individual selection is biased to fitter individuals, giving them a proportionally higher chance of being selected. Reproduction involves two types of genetic manipulation, namely crossover and mutation. The simplest crossover operator is single point where genetic information is swapped after a random position, producing two new off springs. Mutation is another genetic operator that is applied to all new chromosomes with a set probability. In the binary string representation, mutation will cause a random bit to change its state, 0 to 1 or vice versa. Mutation can be considered a background operator that ensures the probability of finding the optimal solution is never zero. Mutation tends to inhibit the possibility of converging to a local, rather than the global optimum. After reproduction, the cycle is repeated. New individuals are decoded and the objective function evaluated to give their fitness values. Individuals are selected for mating according to fitness and so the process continues. The average performance of individuals in a population is expected to increase as good individuals are preserved and bred, while less fit members die out. The GA is terminated under a given criteria, for example, a certain number of generations have been completed, a level of fitness has been obtained or a point in the search space has been reached. There are several parameters to fine-tune in a GA, such as population size and mutation frequency. These parameters can be chosen with experience or through experimentsit.

III. PROPOSED METHODOLOGY

Image processing has shown its importance in modern world. It is a subject of wide research in any field of engineering science. A common image processing task is to apply an image processing algorithm to a series of files. This procedure can be time consuming if the algorithm is computationally intensive, if you are processing a large number of files, or if the files are very large. To make processing faster we use parallel processing.

Here we are showing the task of detecting cells in an image. The algorithm work in steps given below:

1. Reading Image
2. Removing Noise
3. Converting to binary
4. Morphological operations
5. Marking cell area

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 4, November 2012)

The series of this operation is first applied to four images sequentially.
Then all these five operations are arranged in pipelined structure and then all four images marking one to four is processed to that pipeline structure.

GA works on a function known as fitness function. To optimize the our problem we set our goal in terms of time as the mapping sequence which take minimum time to process will be the best. The fitness function starts with taking a random sequence of four images i.e. [3 2 4 1] then it contains the pipelined structure of operations stated above. So, the definition function can be given as

Min Processing Time = Function of {Sequence of Image}

IV. Simulation And Results

All the programs have been developed in MATLAB R2009b using the Parallel Computing Toolbox and Genetic Algorithm Toolbox. Simulation is performed of computer having Intel Core i3 Processor, 2GB RAM, L2 Cache and Windows8.

The Results are as follows:

Processing	Order Of Image Processed	Computational Time
Sequential	[1 2 3 4]	2.1549 Sec
Parallel	[1 2 3 4]	1.5920 Sec
Optimized Parallel	[2 1 3 4]	0.86624 Sec

IV. Conclusions

The paper provided a brief overview of GA. A framework for using GAs to solve Mapping for Parallel Image processing problems was proposed and the results were shown. GAs can be employed to design new and more generic techniques to solve scheduling problems and any advances made in this direction can be extended to other classes of problems that are NP-complete. Also the same problem can be optimized using particle swarm optimization. The results shown for only four images but work can be loaded over the large set of images to achieve faster speed in parallel computation.

References

[1] Wikipedia.com

[2] SOO-YOUNG LEE and J. K. AGGARWAL, "A Mapping Strategy for Parallel Processing" IEEE TRANSACTIONS ON COMPUTERS, VOL. C-36, NO. 4, APRIL 1987

[3] Edwin S. H. Hou, *Member, IEEE*, Nirwan Ansari, *Member, IEEE*, and Hong Ren, A Genetic Algorithm for Multiprocessor Scheduling, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. VOL. 5. NO.2, FEBRUARY 1994

[4] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Mass.: Addison-Wesley, 1989.

[5] J.J. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, Mich.: Univ. of Michigan Press, 1975.

[6] R. Horst and P.M. Pardalos, Handbook of Global Optimization. The Netherlands: Kluwer Academic Publishers, 1995.

[7] H. Ratschek and J. Rokne, New Computer Methods for Global optimizations. Ellis-Horwood, 1988.

[8] C. Guus, E. Boender, and H.E. Romeijn, ªStochastic Methods,Handbook of Global Optimization, R. Horst and P.M. Pardalos, eds.,pp. 829-869. The Netherlands: Kluwer Academic Publishers, 1995.

[9] T.M. Nabhan and A.Y. Zomaya, ªA Parallel Computing Engine for a Class of Time Critical Processes,º IEEE Trans. Systems, Man, and Cybernetics, part B, vol. 27, no. 25, pp. 774-786, 1997.

[10] Handbook of Genetic Algorithms, L. Davis, ed. Van Nostrand Reinhold, 1991.

[11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. New York: Springer-Verlag, 1992.

[12] M. Srinivas and L.M. Patnaik, ªGenetic Algorithms: A Survey,Computer, vol. 27, pp. 17-26, 1994.

[13] J.L. Riberio-Filho, P.C. Treleaven, and C. Alippi, ªGenetic-Algorithm Programming Environments,º Computer, vol. 27,pp. 28-43, 1994.

[14] A.Y. Zomaya, R.C. Lee, and S. Olariu, ªSchedulers that Evolve! Technical Report 96-PCRL-02, Parallel Computing Research Laboratory, Dept. of Electrical and Electronic Eng., The Univ. of Western Australia, 1996.