

# A Framework for Parallel Genetic Algorithms on PC Cluster

Guangzhong Sun, Guoliang Chen  
Department of Computer Science and Technology  
University of Science and Technology of China (USTC)  
Hefei, Anhui 230027, P.R.CHINA

*Abstract:* - Genetic algorithms (GAs) is a kind of powerful method for solving complex problems. Generally, parallel computing is used together with GAs to achieve better execution time and there exist several models for Parallel Genetic Algorithms (PGAs), such as master-slave model and island model. Currently, PC cluster is the common architecture for parallel computing with high performance-price-ratio. In this paper, we choose island model for implement PGA on PC cluster and present a novel framework for PGAs (FPGA-Cluster) using updated island model on PC cluster. A GA program can be translated to a PGA program on PC cluster by FPGA-Cluster easily and efficiently, and an application for FPGA-Cluster is also demonstrated.

*Key-Words:* - Genetic algorithms, Parallel computing, Cluster, Framework, Island model, Master-slave model

## 1 Introduction

Genetic algorithms (GAs) is a kind of efficient search method inspired by genetics of nature and has been applied to find acceptable solutions for complex problems in different fields, such as optimization problems, classification problems etc. <sup>[1]</sup> Even so, the execution time of GAs is often too long for scientific researchers. There are several efforts for this problem and one important method is using the techniques of parallel computing. <sup>[2]</sup>

Parallel genetic algorithms (PGAs) are not just the parallel version of sequential genetic algorithms. <sup>[3]</sup> There are some works discussed PGAs on parallel computers. <sup>[4-7]</sup>

In the last decade, with the rapid improvement of the performance of PC and network, PC cluster became the most popular architecture for parallel computing. At present, PC Clusters are widely deployed and easily to access. <sup>[8]</sup>

Because of the large amount of PC Cluster users, how to implement PGAs easily and efficiently on PC cluster is a valuable research issue. In this paper, we give a novel framework for PGAs on PC Cluster based on updated island model. With this framework, programmer can implement PGAs by using Message Passing Interface (MPI) on PC cluster easily and efficiently and dramatically improvement of abstraction, generality as well as flexibility can be achieved.

The rest of this paper is organized as follows: Section 2 gives a full discussion of FPGA-Cluster with design policies and implementation details; Section 3 shows an example of outlier analysis for gene expression data using FPGA-Cluster. In the last

part, issues about the future work and conclusions are presented based on what we have experienced.

## 2 Design and Implementation of FPGA-Cluster

In this section, the design and implementation policies are discussed in detail. First, we analyze the features of PC cluster. Second, with the comparison between several PGA models, the island model is selected and modified to be used in FPGA-Cluster. At last, the architecture of FPGA-Cluster and the implementation are de-scribed with more details.

### 2.1 Features of PC cluster

PC cluster is a computing system by interconnecting a number of individual PCs. With the rapid development of PC and network, PC cluster becomes one of the most popular parallel computing systems with high performance and appropriate price. With help of a guide, users can easily build such a system by themselves. So for research and application of parallel genetic algorithm, how to implement a PGA on PC clusters is a valuable research issue.

PC cluster has a distributed memory, where each node of the cluster is a personal computer. In such architecture, the communication between different nodes usually is the bottleneck of the whole system. <sup>[9]</sup>

On parallel computing environments based on distributed memory, the message passing model is more intuitive and easy to implement. Typical message-passing libraries include the Parallel Virtual

Machine (PVM) [10] and the Message Passing Interface (MPI) [11]. Both of them can be easily implemented in homogeneous or heterogeneous distributed environments, and so became prevailing. But for common users, they are still too complicated. Recently, many researches have been done for reducing the difficult of parallel programming. [12]

## 2.2 Island Model in FPGA-Cluster

### 2.2.1 Classification of Parallel GAs model

Mainly, there are two types of parallel GAs: one is fine grain GAs and the other is coarse grain GAs. Among them, the master-slave model and the island model are the representative models for each type.

The master-slave model is also called as the farming model. [3] In this model, a central unit, which is the master, controls the computing process of the global population. Other units, which are the slaves, process computing jobs under the control of the master. Each step the master should communicate with the slaves, so this model is a fine grain model. For PC cluster, that communication may be the bottleneck for achieving good performance.

The island model is also called as regional model or multideme model. [5] It consists of a number of subpopulations that compute asynchronously and independently. Each subpopulation runs a sequential GA and they exchange some individuals between each other, which are often called migration. Different from master-slave model, the island model is a coarse grain model. For consistency between computing architecture and algorithm model, we assume the island model is suit for PC cluster. The recently research based on experiments [6] also show that.

### 2.2.2 Updated island model

Here is an update of the traditional island model. Different from the traditional island model, each bulk in BSP model contains several sequential GAs and a global communication at the end of bulk, as shown in figure1. Immigrants are moved by global communication.

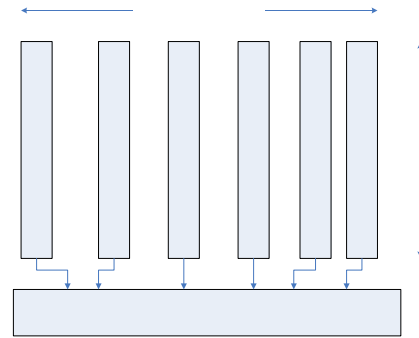


Figure1 Updated island model for PGAs

This update is inspired by BSP (Bulk Synchronous Parallel) model of parallel computing [13]. It balances the communication and computing, and it's suitable for PC cluster systems. This model can also reduce the difficult of design and analysis, and can easily ensure the correctness of algorithm.

In this new model, some new parameters should be added as follow.

- Number of subpopulations
- Migration frequency
- Number of migrations
- How to select migrants
- Which individuals to replace by immigrants

Once the above parameters are all decided, the PGA can be decided based on the sequential GA easily and it implies we can create a tool to finish the parallelization process automatically.

## 2.3 Architecture of FPGA-Cluster

### 2.3.1 I/O of FPGA-Cluster

The input and output of FPGA-Cluster is shown in Figure2.

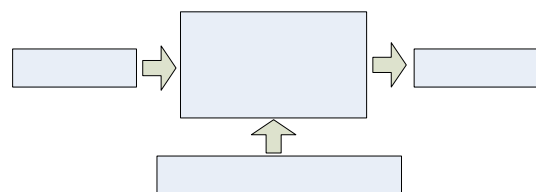


Figure2 input and output of FPGA-Cluster

*Sequential GA* is written in C language with additive compiler directive. The compiler directive points out the data type of individual's representation.

New parameters are included, such as number of subpopulations, migration frequency, number of migrations, how to select migrants and which individuals to replace by immigrants and all the parameters prepared in a configuration file. Here we give the definition of all the new parameters.

- *Number of subpopulations*: an integer, which is the number of parallel processes.
- *Migration frequency*: an integer, which is the CPU computing time of local process. If one process finished the computing and others didn't, it should wait until all process has reached the point.
- *Number of migration*: a floating point numbers between 0 and 1, which is the percent of individuals' numbers in sequential GA.
- *How to select migrants*: an integer. In FPGA-Cluster, there are two kinds of methods to select migrants. One is best first, the other is random choose. We use 0 and 1 to mark them, respectively
- *Which individuals to replace by immigrants*: also an integer. Just like the strategy how to select migrants

Parallel GA is built with MPI C binding. Here we choose LAM MPI [14], which is a popular MPI library.

### 2.3.2 Implement of FPGA-Cluster

The key issue of the implement of FPGA-Cluster is the compiling technology, which convert the sequential codes to parallel code with MPI. That is, FPGA-Cluster can be view as a compiler or auto-parallelization tool. [15]

- Computing

The computing part of final parallel program is come from the corresponding part of the sequential program, such as evaluation, selection and genetic operations (including mutation, crossover, etc.)

- Communication

The communication part of final parallel program is generated automatically based on the compiler directive in sequential program and the new parameters in configuration file. In FPGA-Cluster, MPI functions used in final parallel program is shown below:

int **MPI\_Alltoall**(void\* sendbuf, int sendcount, MPI\_Datatype sendtype, void\* recebuf, int recvcount, MPI\_Datatype recvtype, MPI\_Comm comm): all to all communication function, which is used to exchange the migrations.

int **MPI\_Barrier**(MPI\_Comm comm): barrier function, which is used to make all processes synchronous.

int **MPI\_Bcast**(void\* buffer, int count, MPI\_Datatype datatype, int root, MPI\_Comm comm): broadcast function, which is used to report the result to all processes when one can a accepted solution.

The data structure of migrations is defined in sequential program, which are added as compiler directive.

- Input and Output

Since parallel programs written by MPI C binding are SPMD (Single Program Multiple Data style. In it, the input and output of the final parallel program should be processed by only one process, which is implemented by assigning a fixed process for I/O.

## 3 Case Study: Implementing a PGA with FPGA-Cluster

In this section, we choose a parallel algorithm to be parallelized with FPGA-Cluster. The process of applying FPGA-Cluster is described in detail after the list of the corresponding sequential GA. The experiment results of the PGA are also presented.

### 3.1 A Sequential GA

Here we use the sequential GA for outlier analysis for gene expression data in [17], which is written with C language. In it, the individuals are coded as d-dimensional vector, which is described as a float array with length of d.

### 3.2 Application of FPGA-Cluster and Experiment Results

To use FPGA-Cluster, we should decide the new parameters. For the selected GA program, we choose the following values as table1 shown.

Number of subpopulations	8
Migration frequency	100s
How to select migrants	Random
Which to replace by immigrants	Best first
Number of migration	10%

Table1 parameters' values for PGA program

For FPGA-Cluster, the value of parameters can be changed easily. Once we modify them in configuration files, a new PGA program will be

generated after compiling. In the parallel program, the computing part is just as sequential program. The communication part is implemented by MPI functions.

The performance of PGA program is tested on an 8-node PC Cluster connected with Ethernet. The input data are from [17], which is two lymphoma expression matrixes. The experiment results are shown in the table2 and table3.

M	Sequential GA (s)	PGA (s)
300	3308	646
400	4179	670
500	3277	662
600	3564	689
700	3845	673

Table2 Test on Lymphoma Data ( Blood B cells)

M	Sequential GA (s)	PGA (s)
300	3779	656
400	4095	680
500	3684	629
600	3573	658
700	3734	637

Table3 Test on Lymphoma Data ( Blood B cells; memory CD 27+)

Both sequential and parallel GA programs can get the best solutions and the execution time is reduced obviously with parallel program. The results also show that the speedup does not achieve the number of processors because of the communication cost. However, it proves the parallel computing can reduce the execution time of the programs.

#### 4 Conclusion and Future Work

In this paper, a framework for parallel genetic algorithms on PC cluster (FPGA-Cluster) is presented. FPGA-Cluster makes use of parallel programming and island model to implement parallel genetic algorithm. With the help of FPGA-Cluster, programmers can turn genetic algorithms into parallel algorithms intuitively, rapidly and effectively. FPGA-Cluster can also help the design and analysis of parallel genetic algorithms. The advantages are shown below:

*Abstraction* Programmers using FPGA-Cluster are able to pay more attention on the algorithms and overall structure of parallel programs, instead of running into the boring and error-prone implementation details of parallel programming.

*Efficiency* In virtue of suitable between island model and PC cluster, there is almost no overhead introduced by abstraction. Parallel programs written in FPGA-Cluster have a good performance.

The future work on FPGA-Cluster includes:

- To make graphical user interface (GUI) for FPGA-Cluster
- To use more parallel genetic algorithm models, besides island model

Otherwise, the final version of FPGA will include FPGA-Cluster and FPGA-SMP for different parallel computer architectures.

#### Acknowledgements

This work is supported by the National Science Foundation of China under the grant No.60533020.

#### References:

- [1] David Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, (1989) .
- [2] Cantú-Paz, E, Efficient and Accurate Parallel Genetic Algorithms. Boston, MA: Kluwer Academic Publishers. (2000)
- [3] Erick Cantú-Paz: A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*. Vol. 10, No. 2. (1998) 141-171.
- [4] Mariusz Nowostawski, Riccardo Poli: Parallel Genetic Algorithm Taxonomy. (1999)
- [5] Sven E. Eklund: A massively parallel architecture for distributed genetic algorithms. *Parallel Computing*, 30. (2004) 647-676.
- [6] Jason G. Digalakis, K. G. Margaritis: Real Coded Parallel Memetic Algorithms on Messages Passing Clusters, <http://macedonia.uom.gr/~jason/jason2005bulgaria.pdf>, (2005)
- [7] Cantú-Paz, E. Markov chain models of parallel genetic algorithms. *IEEE Transactions on Evolutionary Computation*. 4(3), (2000), 216-226.
- [8] Top500 high performance computers, <http://www.top500.org/>
- [9] Guoliang Chen: Parallel Computing — Structure, Algorithm and Programming. High Education Press. (2003) 57-75.
- [10] PVM: Parallel Virtual Machine. [http://www.csm.ornl.gov/pvm/pvm home.html](http://www.csm.ornl.gov/pvm/pvm%20home.html).
- [11] Message Passing Interface Forum: MPI: a message-passing interface standard. *International Journal of Supercomputer Applications*, 8(3/4), (1994)

- [12] Zhen Yao, Qilong Zheng, Guoliang Chen: GOOMPI: A Generic Object Oriented Message Passing Interface. IFIP International Conference on Network and Parallel Computing. (2004) 261-271.
- [13] L. G. Valiant, A Bridging Model for Parallel Computation, Communication of ACM, 33(8), (1990) 103-111.
- [14] LAM MPI. <http://www.lam-mpi.org/>
- [15] A. V. Aho, R. Sethi, J. D. Ullman: Compilers: Principles, Techniques and Tools, Addison-Wesley, (1986).
- [16] Chao Yan, Guo-liang Chen, Yi-fei Shen: Outlier Analysis for Gene Expression Data. Journal of Computer Science and Technology, 19(1), (2004) 13-21.
- [17] <http://lmpp.nih.gov/lymphoma/>.