The System for Evolutionary Computing on the Computational Grid

Yusuke Tanimura
Graduate School of Engineering
Doshisha University
Tatara Miyakodani 1-3, Kyotanabe, Kyoto, Japan
email: tanisuke@mikilab.doshisha.ac.jp

Tomoyuki Hiroyasu
Department of Engineering
Doshisha University
Tatara Miyakodani 1-3, Kyotanabe, Kyoto, Japan
email: tomo@is.doshisha.ac.jp

Mitsunori Miki
Department of Engineering
Doshisha University
email: mmiki@is.doshisha.ac.jp

Keiko Aoi
Graduate School of Engineering
Doshisha University
email: aoi@mikilab.doshisha.ac.jp

ABSTRACT
The computational Grid can offer users tremendous computer resources. Many researchers are developing the Grid middleware and the typical results are Grid RPCs. However, application models are restricted to use when Grid RPCs are applied. In this paper, we proposed the "EVOLVE/G" system for developer to construct evolutionary computation (EC) system on the computational grid. The EVOLVE/G has a tree topology of data communication. In the EVOLVE/G, there are Agent and some Workers. Since the data can be transferred between Agent and Worker, any logical model of EC can be implemented by the EVOLVE/G. Furthermore, it has mechanism of clustering nodes. Therefore, the effective model can be constructed on the Grid environment. In this paper, the Grid calculation model of Parallel Simulated Annealing using the Genetic Crossover (PSA/GAc) is built using the EVOLVE/G and presented the experimental results in the real Grid environment. Consequently, it is shown that the examination of the calculation model using the EVOLVE/G is effective.

KEY WORDS
Grid Computing, Evolutionary Computing, Optimization

1 Introduction

The Grid technology enables to integrate the computational resources and information resources that exist in a wide area. Using these integrated resources, we can perform distributed / parallel computing in a wide area. Therefore, it is expected as technology which solves the demand of the large-scale computation[1]. Especially, the recent research of the Grid contributes the constructing the Grid middleware and many testbeds are constructed and running. They are served as a stage of examining application. The Grid RPC based system is a typical middleware to use the Grid environment[2][3].

Our goal is to prepare tools for users, who want to develop optimization methods by evolutionary computation (EC) on the computational Grid. Since it needs a high calculation cost to solve optimization problems such as structural optimization problems, job shop scheduling problems, protein tertiary structure problems and so on, huge computational resources are needed. One of the characteristics of EC is multi point search. Because of this characteristic, there are several ways to perform EC in parallel. For example, in the genetic algorithm (GA), master-slave model[4], distributed population model[5], and cellular model[6] are parallel models.

When a parallel model is used, it is expected that EC can be applied to parallel computers and it is also expected that the searching ability becomes increased. Therefore, developers want to apply their parallel models of EC even in the computational Grid. However, when they use the Grid RPCs as the developing tools, only a master-slave model (Figure1(a)) can be applied. Developers sometimes use distributed population model (Figure1(b)). To construct this system, developers need to use raw socket communications or Globus Toolkit. However, this development cost is very high.

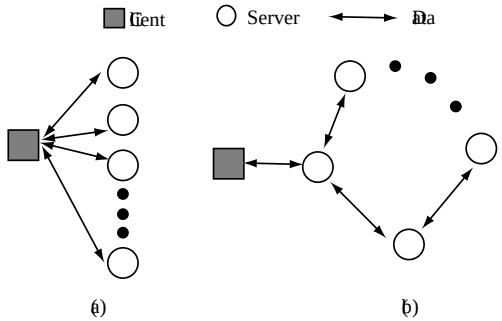

Figure 1. Master-slave model and distributed population model of EC

In this paper, we propose a new tool for developing systems of EC in the computational Grid. That is called the EVOLVE/G. Using the EVOLVE/G, developers can apply their models of EC freely. At the same time, the EVOLVE/G has the mechanism to classify into sub groups

according to CPU power, I/O, network ability and so on. In this clustering, developers simply define the clustering condition, but they do not define the node nor the machine.

To discuss the effectiveness of the EVOLVE/G, the system of the PSA/GAc[7] which is one of the evolutionary computation methods is developed. This developed system is applied to solve protein tertiary structural problem. Through this experiment, we discuss the usefulness of the EVOLVE/G and the Grid model of the PSA/GAc.

2 Grid Oriented Computing Application

There are some common features in the application that can use the Grid environment effectively. In this paper, the application that is satisfied with those features is called GOCA (Grid Oriented Computing Application) and defined it as follows.

- The task can be divided into several sub tasks.
- The sub task has few dependencies for other sub tasks or it can be executed independently.
- The sub tasks can be executed in parallel.
- The communication between sub tasks is not need at all or is not need frequently.
- Low cost for continuing the job when some nodes are removed.
- The job will utilize new nodes added to the system.

The evolutionary computation application is satisfied with the conditions of GOCA. However, in the existing Grid RPC based system, it is difficult to bring out the features of GOCA to the maximum. Then, in this research, the EVOLVE/G is developed as the Grid middleware which can harness the feature of GOCA.

3 The EVOLVE/G System

The EVOLVE/G is a grid middleware to enable easy implementation of the various evolutionary computing in GOCA. Figure2 shows the basic architecture of the EVOLVE/G.

The EVOLVE/G uses Globus 1.1.4 and it is described in C and Java languages. The EVOLVE/G prepares API for the application developer as an interface to implement each component of Worker, Agent and Super Agent. These API are similar to Send() and Recv() function of MPI. Using these APIs, application developer can describe data flow and management freely. The EVOLVE/G consists of Agent and Worker. Agent checks Worker every fixed time and gives a command to Worker, acquires the progress, the state of the node or results. Worker performs under management of Agent and each Worker is not concerned where other worker performs. Therefore, developers implement the behavior of Agent and Worker. It specifies what data from one Worker is transmitted to other workers at what timing.

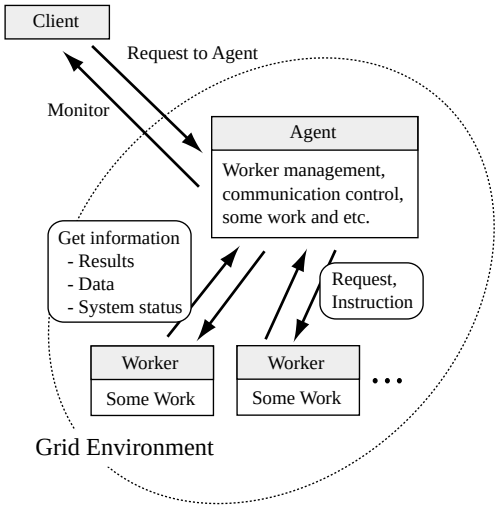

Figure 2. Basic architecture of the EVOLVE/G

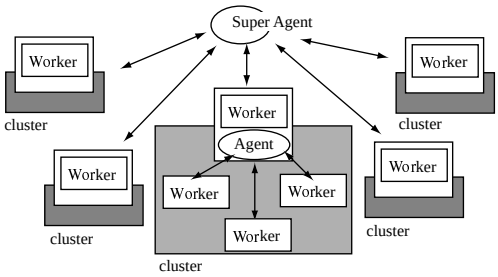

Figure 3. Basic architecture in hierarchical topology of the EVOLVE/G

The topology of data communication in the EVOLVE/G is a tree topology. However, since data is communicated between Agent and Workers, any logical model of EC can be implemented.

The EVOLVE/G can also have a hierarchical topology of data communication. This topology is constructed by clustering the nodes which can be used within the Grid environment based on the machine information or network information connected with them. In this case, there is Agent in a cluster and this Agent is also Worker for the upper level. This worker is communicated with Super Agent. Again, this tree structure is a topology for data communication and the logical model of EC can be different from this. When there are two levels, developers have to implement the behavior of Agent and Worker in the upper level and lower level. At the same time, developers have to define the rule of clustering. This classification is based on the performance of the node, the throughput of the network or the site that belongs. The concept is summarized in Figure3.

It is planned that the start of Agent and Worker is performed automatically by another system. At this point, the clustering of nodes is executed. In the following simulation, this operation is performed by hand.

4 Parallel Simulated Annealing Using Genetic Crossover

The Parallel Simulated Annealing using the Genetic Crossover (PSA/GAc)[7] is a hybrid algorithm. In the PSA/GAc, the SA processes are executing in parallel, while the information about search points is sometimes exchanged by the genetic crossover.

Generally, the SA and GA are said to have the following characteristics. SA is a generic approximation method which tries to solve optimization problems through simulated annealing process. An annealing process is a physical process of gradually cooling a molded material at the high temperature to generate a low energy state such as crystals of a few defects. SA searches repeating three processes. Generate process creates the following state from the present state. Accept criterion process judges whether it changes to the following state. Cooling process generates the following temperature from the present temperature. Clearly, SA has only the present state and searches the optimal solution with emphasis on the near.

On the other hand, the GA is an engineering algorithm imitating evolution and selection of a living thing. In the GA, the individuals only which adapted for environment can survive the next generation. The GA has two or more individuals as a solution candidate and can perform the large region-search. However, in many cases, crossover operator that generates the solution candidate of the following state from combination of two searching points, cannot search well for near the search point.

The PSA/GAc is an algorithm devised to utilize these merits. It performs the partial search by the PSA and large region-search by the crossover operator of the GA.

The search procedure of the PSA/GAc is shown in Figure4.

step 1 An initial searching point is generated and several search process of SA runs in parallel.

step 2 When the annealing reaches the fixed cycle d, the pair is generated randomly from parallel SA.

step 3 It performs the one point crossover between design variables on each pair and two children are generated.

step 4 In each pair, two individuals having high evaluation value are selected from 4 individuals, which are two parents and two children.

step 5 It performs annealing process in the fixed cycle d with selected individuals.

step6 Processing from step2 to step5 is repeated until it satisfies the terminal criterion.

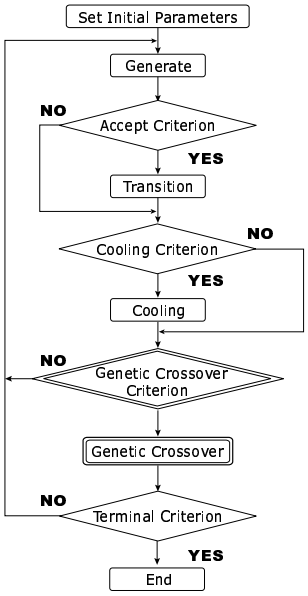

Figure 4. Flow of PSA/GAc calculation

5 Two Logical Models and Implementation of the PSA/GAc Using the EVOLVE/G

This paper examines the Grid computing model of the PSA/GAc using the EVOLVE/G. Here, we develop two models; the basic model and hybrid model. The basic model is implemented by one Agent and the hybrid model is implemented by two types of Agent and two stages.

5.1 Basic model

In the general model, a simple the PSA/GAc is performed. There are several processes of SAs and they are running independently. After some steps, two processes are chosen randomly and new search points are generated by the operation of the genetic crossover with two search points of these processes. After the genetic crossover, operations of SAs are restarted. This logical model in shown in Figure5.

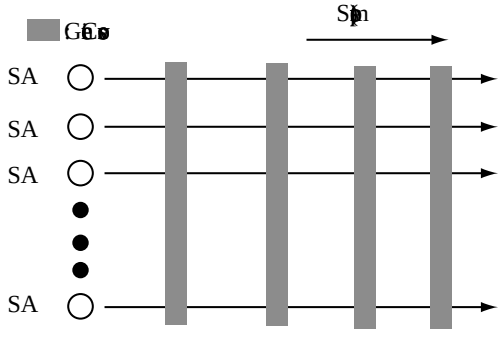

Figure 5. Basic model

This logical model is implemented by one Agent in the Grid and all remainder serves as Worker managed by

the Agent. Each Worker performs SA for some steps. When it ends, Worker writes out the best search point (individuals attending the crossover) to the file and goes into a wait state. For that time, Agent checks each worker and the crossover individuals are sent to Agent. After the crossover and selection on Agent, survival individual is sent from Agent and the genetic crossover operator is completed. After the operation, SA is performed until the next crossover cycle.

Agent checks each worker every fixed cycle specified by the user and realizes the genetic crossover between workers. However, on the Grid environment, the performance of each node running workers is not uniform and it is expected that Agent cannot contact to any Workers according to some troubles. Then, Agent will perform the crossover with pairs, which is generated by the individuals gathered from workers reaching the crossover cycle and enabling communication at the checkpoint.

5.2 Hybrid model

The other logical model is the hybrid model. In this model, there are several the PSA/GAc operations. In the PSA/GAc operation, genetic crossovers are performed. After some steps, the best search point is chosen from each the PSA/GAc. These points are exchanged between the PSA/GAc. This logical model in shown in Figure6.

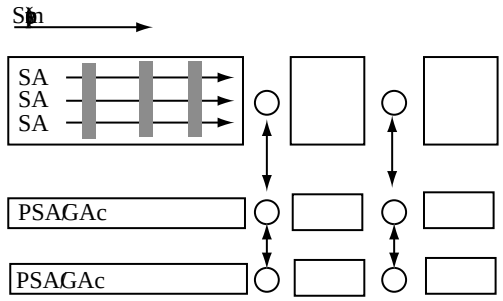

Figure 6. Hybrid model

To build this system, one Super Agent and several Agents are prepared. Each Agent manages several Workers. In this model, the PSA/GAc is performed in a sub cluster. The several sequential SAs are performed on Worker and the crossover is performed through Agent. In the upper cluster, Super Agent monitors the searching progress of each sub cluster and exchanges the best search point among them. One Agent sends the best search point to the Super Agent. Other Agent receives it through Super Agent.

In this implementation, it is needed to define how the clusters are constructed. In the following simulation, our classification is based on that PC cluster systems the node belongs. In the simulation, we use the Grid made from 3 PC cluster systems. Therefore, three clusters exist in the simulation. However, important thing is the developer does not know how many clusters create. That means, if there are 5 PC cluster systems in the simulation, five clusters will exist.

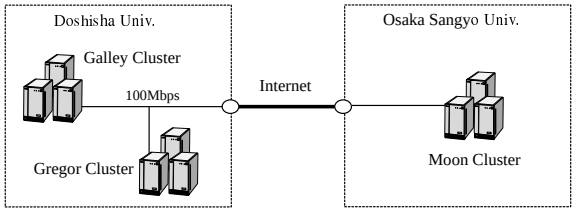

Figure 7. Network environment

Table 2. Thrughput

Network	Thruput	
	Message size (1KB)	Message size (1MB)
Galley–Gregor	10.95 [MB/sec]	11.21 [MB/sec]
Galley–Moon	107.7 [KB/sec]	92.12 [KB/sec]

6 Experimental Results

6.1 Machine and Networks

Two PC clusters installed in Intelligent Information Center at Doshisha University and the PC Clusters installed in Osaka Sangyo University were used for our experiment. The specification of each cluster is shown in Table1. Linux is used as OS of each node. The Globus Toolkit and MPICH are installed in the gateway machine. On all nodes, Java runtime environment is usable. Figure7 shows that Galley and Gregor cluster installed in the same building are connected with 100 Mbps networks, but the networks between Galley and Moon, between Gregor and Moon are connected through the Internet. The network thrughput between the gateway machines before and after our experiment is shown in Table2.

In our experiment, Agent or Super Agent is executed on the gateway machine on Galley and it spawns other Workers. Spawned Worker or Agent is executed on other 8 nodes on Galley, 12 nodes on Gregor and 4 nodes on Moon. Agent and Worker is executed on the same nodes. A total of 24 Workers will be executed on each node.

6.2 Optimization Problems

In our experiment, we solved the test problem of minimizing the energy function to predict the protein tertiary structures, which function was proposed by Okamoto[8]. Target protein was Met-Enkephalin that is very small-scale protein and consists of 5 amino acid residues of Tyr-Gly-Gly-Phe-Met. It is said that this protein has the minimum energy structures in the range of $E \leq -11kcal/mol$ inside of the gaseous field based on the ECEPP/2 energy function[8]. In

Table 1. Machine Specification

Site	Cluster name	System
Doshisha Univ.	Galley	Pentium III (1.1GHz), 1CPU Pentium III (850MHz), 8CPU
Doshisha Univ.	Gregor	Pentium III (1GHz), 64CPU
Osaka Sangyo Univ.	Moon	Pentium 4 (1.7GHz), 8CPU
Installed Software		
Globus Toolkit 1.1.4, Sun JDK 1.4.0, MPICH 1.2.3		

Table 3. Time costs of calculating the energy function

Cluster	1 Basic model	Hybrid model
Galley	0.215	41.4
Gregor	0.173	33.2
Moon	0.109	20.9

Table 4. Parameter setting of PSA/GAc

Number of SAs on each node	16
Initial temperature	2.0 (1000K)
Last temperature	0.10 (50K)
Crossover cycle	192 step
Range size	$180° \rightarrow (180 \times 0.3)°$

our experiment, the design variables are 10 dihedral angles of the main chain $\phi_1, \psi_1 \sim \phi_5, \psi_5$ and 9 dihedral angles of the side chain $\chi_1^1 \sim \chi_5^4$ at Met-Enkephalin.

Table3 shows the calculation time of the sequential SA for this problem. These results are different on the node of each cluster. Table3 shows the result of executing 1 step on 16 sequential SAs and executing 192 steps.

6.3 Results

The parameter setting in our experiment is shown in Table4. In the basic model, PSA/GAc is performed using 24 nodes. In the hybrid model, PSA/GAc using 8 nodes, 12 nodes and 4 nodes is executed in parallel.

Firstly, the searching progress in the basic model is shown in Figure8. Figure8 shows the energy value of the best search point and the number of step when the point is gathered by Agent at the checkpoint. These results show that the searching process of Workers in one PC cluster system is separated from the one of Workers in another PC cluster system. This also shows that there is a possibility for promotion of search efficiency and improvement in an execution performance by clustering.

Secondly, the result of comparison of the general model and hybrid model is shown in Figure9 and Figure10. In Figure9, the horizontal axis is the number of steps. In this graph, Agent acquires the searching point having the

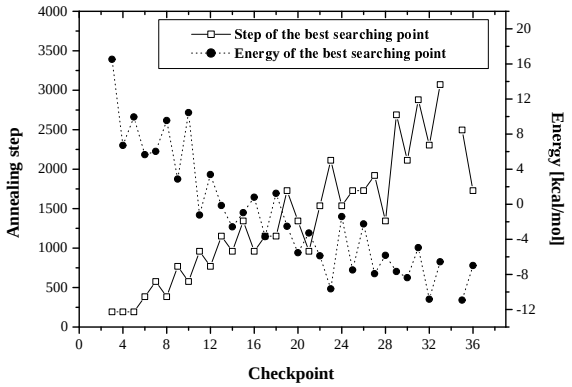

Figure 8. Progress of the protein energy and search step

same step as before at the different checkpoint, even though the performance of the node running Workers is different. This figure shows that the hybrid model can find the better solution than the basic model.

On the other hand, Figure10 also shows the results of the comparison of the basic model and hybrid model. However, in this figure, the horizontal axis is elapsed time. In the basic model, Agent acquires information from the group of early search Worker or the group of late search Worker at each checkpoint. Then, the best search point gathered from Workers may be good or bad along the time line. This figure also shows that the hybrid model performs the effective search compared with the basic model by the viewpoint of the time line.

6.4 Discussion

The hybrid model has two advantages. Firstly, it is useful for not-scaled application. Performing the parameter sweep of parallel executing is more effective for the application that has a not-scaled algorithm. In the example of the PSA/GAc, 8 or 12 parallel execution might show the better search than 24 parallel execution. This is our future research subject. In addition, evolutionary computation enables not only performing the simple parameter sweep but also exchanging some information between clusters and tuning the more suitable parameter on each cluster.

Secondly, the scalability of the performance is obtained by making the hybrid model. The EVOLVE/G has a

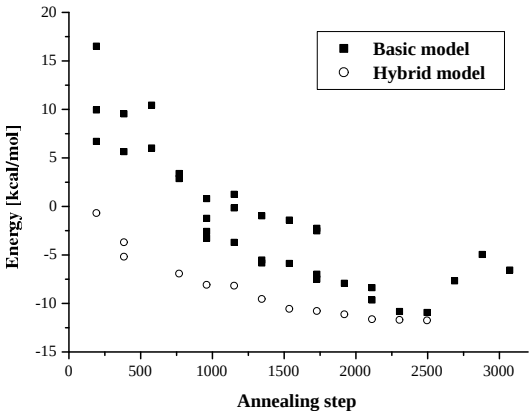

Figure 9. Comparison of basic model and hybrid model (Viewpoint of step)

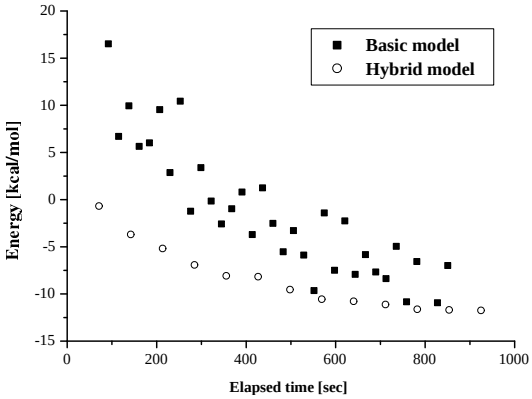

Figure 10. Comparison of basic model and hybrid model (Viewpoint of time)

hierarchical topology for data communication. On this system, any logical model can be constructed. In the hybrid model proposed in this paper, it performs frequent communication for the genetic crossover over Workers in the lower cluster. It is an effective model where nodes are classified into sub groups according to the network distance. By this clustering, the fine grained communication is occurred in a cluster system and the coarse grained communication is performed over the cluster systems.

The EVOLVE/G system enables clustering. Then, any logical model such as the hybrid model can be constructed. It is very useful as the Grid System.

7 Conclusion

In this paper, the EVOLVE/G system, which is a Grid tool for developer of evolutionary computation, is proposed. This system consists of Agent and multiple Workers. Since the data can be exchanged between Agent and Workers freely, any logical models of EC can be integrated. This system also has the mechanism of clustering nodes on the Grid. These clusters are placed in the tree topology.

Using the EVOLVE/G, the Grid model of the PSA/GAc is implemented which is one of the applications of EC. Two types of logical models of the PSA/GAc are prepared; the general model and hybrid model. For the hybrid model, the clustering of the nodes that are based on the distance of the network has been performed. In the simulation, the protein tertiary structure problem is solved. The experiment is performed in the real Grid computing environment. Through the experiment, it is shown that the hybrid model has a good performance. In the hybrid model, the fine grained communication is performed in a PC cluster and the coarse grained communication is occurred between PC clusters. As a result, it presents that the EVOLVE/G system is useful to develop systems of evolutionary computation.

Acknowledgments

This work was supported by a grant to RCAST at Doshisha University from the Ministry of Education, Science.

References

[1] I.Foster and C.Kesseleman, *The Grid: Blueprint for a New Computing Infrastructure* (San Francisco: Morgan Kaufmann, 1998).

[2] H.Casanova and J.Dongarra, Netsolve: A Network Server for Solving Computational Science Problems, *Int. J. of Supercomputer Applications and High Performance Computing*, Vol.11, No.3, 1997, 212-223.

[3] H.Nakada, M.Sato and S.Sekiguchi, Design and Implementations of Ninf: toward a global computing infrastructure, *Proc. of Future Generation Computing Systems, Metacomputing Issue*, Vol.15, 1999, 649-658.

[4] D.Levine, A parallel genetic algorithm for the set partitioning problem, *T.R. No. ANL-94/23* (Algonne National Laboratory, Mathematics and Computer Science Division, 1994).

[5] R.Tanese, Distributed genetic algorithms, *Proc. of the 3rd ICGA*, 1989, 434-439.

[6] M.Gorges-Schleuter, ASPARAGOS: an asynchronous parallel genetic optimization strategy, *Proc. of the 3rd ICGA*, 1989, 422-428.

[7] T.Hiroyasu, M.Miki and M.Ogura, Parallel Simulated Annealing using Genetic Crossover, *Proc. IASTED PDCS 2000*, Las Vegas, USA, 2000, 139-144.

[8] Y.Okamoto, T.Kikuchi and H.Kawai, Prediction of Law-Energy Structures of Met-Enkephalin by Monte Carlo Simulated Annealing, *CHEMISTRY LETTERS*, 1992, 1275-1278.

[Source]

14th International Conference on Parallel Distributed Computing and Systems (PDCS 2002), pp.39–44, Nov. 4–6, 2002.