Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 4, No. 4, November 2013), Pages: 1-7
www.jacr.iausari.ac.ir

# Presentation of a Comparative Method for Grid Reliability

**Sedigheh Navaezadeh[1]✉, Iman zangeneh[2], Mehnoosh vahebi[3]**

*(1) Sama Technical and Vocational Training College, Islamic Azad University, Mahshahr, Branch Mahshahr, Iran*

*(2) Department of Computer Engineering, Ramhormoz Branch, Islamic Azad University, Ramhormoz, Iran*

*(3) Sama Technical and Vocational Training College, Islamic Azad University, Ahvaz Branch, Ahvaz, Iran*

snavaezade@yahoo.com; zangeneh_i@yahoo.com; M_vahebi880@yahoo.com

**Received: 2013/06/12; Accepted: 2013/09/03**

## Abstract

*Every year, with regard to rapid development of data, Grid computing that is a kind of distributed computing system is that it has attracted the attention of most people and considerably taken into account. Computing that cannot be done by huge computers can be performed by Grid computing. If load balance is used, efficiency will increase in Grid. Resources have an important and effective role in Grid efficiency. When resources are available and accessible, the access time and efficiency will improve. In this paper, swarm intelligence algorithm has been used and applied in Grid in order to provide load balance. Since Grid environment is a dynamic environment, glowworm algorithm is used to provide load balance in sites. The results of algorithm simulation presented by Matlab software show that, by using this algorithm, efficiency as well as reliability will increase in Grid environment. Also, this algorithm has been compared with two other algorithms in this field.*

***Keywords:*** *Swarm Intelligence Algorithm, Glowworm Algorithm, PSO-LA(learning automat), Grid, Load balance*

## 1. Introduction

In the past, one of the greatest problems of those who required long and complicated computations was lack of suitable software and hardware facilities to do computations in a proper and logical time. Due to this issue, sometimes, computing continues for several days, months or years. It was observed that, after long computing, an error occurred in the code of the program; therefore, the fault was found in whole computing. Later, in order to solve this problem, the huge computer was used. This computer had high processing power, and could compute fast, so less time was required.

There are two limitations in the use of this huge computer:

A) The power of the processor: whatever the computer is very powerful; its speed does not exceed its own limit due to the limitations existing in the technology.

B) Financial constraints: this computer is very costly, and it cannot be bought and used by many organs.

Therefore, the researchers and scientists of computer science tried to solve this problem. They considered Grid and used it. Grid was proposed by Foster and Classman in the late 90s in order to provide easy availability [1].

Grid computing systems have been created as an important and new field. Also, they are different from distributed systems, and this is due to resource subscription in large scales and innovative application programs with high efficiency. Grid computing systems are a kind of distributed system involving wide range. The main and specific problem of Grid concept is as follows: coordination of resource subscription, dynamic problem solving, and multi-distributed programs. Subscription with which Grid computing is involved is not considered as the primary exchange of the file, but it is direct access to computers, software data and other resources, and these issues and problems require broad spectra of the strategies in terms of problem solving and resource subscription. Therefore, reliability of Grid computing is influenced by the relation existing between the important resources and computing programs [9].

Since scheduling is considered as NP-hard issue, and management of its resources and their scheduling have important roles in load balance, they play effective roles in Grid efficiency [2]. The advantages of Grid computing are as follows: using the resources, the ability of parallel computing, providing resources and virtual organization, reliability and availability of more and extra resources [3].

Since artificial intelligence algorithms are efficient in optimization, they are considered as a good option for some problems in distributed system such as load balance. The aim of load balance is to decrease the difference of load rate existing between the heaviest node and the lightest node in terms of load rate [4].

## 2. Literature Review

The algorithm of load balance is divided into various groups. This algorithm is divided into two groups; namely, static and dynamic. The difference of load balance in static and dynamic mode is that, in static mode, the decisions related to load balance are made in compiling time, while, in dynamic mode, decisions related to load balance are made in running time. It means that, in static mode, these decisions are made in the time of request for the resource, but, in dynamic mode, the behavior of balancer varies according to the changes occurring in parameters and policies.

Load balancing is divided into three groups: centralized, non – centralized and hierarchical groups. In centralized method, all functions are scheduled by a scheduler. In non- centralized method, there is no central scheduler, and scheduling operation is performed by the applicant resources. In hierarchical method, scheduling is organized in the hierarchical form [6].

In load balance of centralized method, many studies have been carried out. Genetic algorithm has been used and applied by [5] in order to balance the load, and its Simulation results have been compared with the algorithm of Min – Max and Max – Min. A new genetic algorithm has been propos by [2]. In this algorithm, resource fault occurrence history (RFOH) has been considered for certain and reliable scheduling in Grid computing. This strategy saves source fault occurrence history in Gird information server. One of the swarm intelligence algorithms of particles is PSO considered as a solution for optimization in terms of quest spaces along with the modeling of social behavior when there are some objectives [7].

A new model on the basis of PSO is PSO-LA in which the learning automata have been used to arrange the behavior of particles and to provide the balance between local and general quest. The results of these experiments showed that this plan performs better and more suitable than the model of standard PSO. In order to solve the problem of local optimization, the algorithm of particles collective movement has been proposed. Gaussian jump in standard PSO as well as some of the changed versions has been used. Gaussian distribution has been used in jumping. In the algorithm of particles collective movement, total current speed of the particle is computed in order to compute the speed of particle in the next step. In fact, the speed of particle involves two parts in each step. The first step is related to finding the best personal experience and the best group experience. Without the second part, algorithm is in the form of a blindly general quest, and without the first step, algorithm is in the form of the local quest in the near of the best particle, and it will be unable to reach the quest space. Through combining these two parts, the algorithm of particles collective movement provides the balance between local and general quest [8]. The learner automata are a machine that is able to perform some finite actions. Each selected action is evaluated by the probable environment. The evaluated result is sent to the learner automata in the form of positive or negative signal. Automata apply this response when it selects the next action. In this way, it selects the action receiving reward from the environment as much as possible. On the other hand, automata learn the action that receives more rewards from the environment [6]. One of the advantages of glowworm algorithm, in comparison to other algorithms of swarm intelligence, is that it can be easily implemented. Firefly Algorithm is a type of algorithm obtained from nature and swarm intelligence algorithm, and has been proposed by yang in 2008. This algorithm is a modern technique based on collective behaviors and has been inspired from the social behaviors of the firefly in the nature. Swarm intelligence is a type of artificial intelligence based on collective behaviors, and is based on neutralized and self-organized foundations. Fireflies generate rhythmic and short beams. Optical patterns of each firefly are different from others. Fireflies use these beams for two reasons: pair's attraction process, and attracting hunt. Moreover, these beams are used as a protective mechanism for fireflies. Due to rhythmic beams and rate of radiation and interval rate between light signals, two genders attract each other. Each particle of the firefly in multidimensional quest space is updated by dynamic absorption and based on the knowledge of firefly and its neighbors. Firefly optimization algorithm can be stated as follows:

* All fireflies are single- gender, and regardless of their gender, the factor of pairs' attractiveness has been considered.

* Firefly x attracts all fireflies, and is attracted by all fireflies.

* Attractiveness is related to their glow, so in a pair of firefly, a worm with less light is attracted toward a worm with more light. Attractiveness power is related to their beams, and the light intensity decreases when the distance between two fireflies increases. If a firefly is not brighter than others, their movement will be performed randomly.

* Brighter firefly moves randomly (all fireflies cannot attract them).

* Firefly brightness is determined by objective function value. Light intensity can be easily determined by target function.

* Firefly particles are randomly distributed in quest space according to above principles. There are two main parts in firefly algorithm: attracting firefly and moving toward the attracted firefly.

### *2.1 General Form Of Firefly Algorithm*

As it can be observed in this Figurer, primary coordination and light intensity rate and the distance between firefly particles are firstly determined in quest area. In Quest procedure of firefly algorithm, each firefly is individually compared with other fireflies. If the firefly has less light than the compared firefly, it will move toward the firefly with more light (the problem of finding maximum point), and due to this process, particles are centralized around a particle with more light. In the next generation of algorithm, if there is a particle with more light, then the particles will again move toward the particle with more light. Quest stages must be generated according to maximum number of repetitions. In this paper, in order to optimize the problem of load balance through using glow-worm group intelligence algorithm, a solution has been presented. In this method, each node in the network is considered as a glow-worm. Each glow-worm tries to optimize its own existing load rate, and this work can be done by exchanging the load with other nodes.

## 3. The Proposed Algorithm

The application of glowworm algorithm has been explained in previous sections. In the algorithm presented on the basis of swarm intelligence, all nodes in Grid system are considered as a solution for finding the most optimized mode.

The place (the site) of each node has been shown by existing rate of light. In order to determine the attraction parameter for each node, its node light rate is compared with neighbor nodes. Each node always moves toward the best neighbor. This work can be done by attracting toward the neighbor or emitting more light from neighbor nodes when the rate of a node light is as same as the rate of neighbor nodes. The movement is performed randomly.

Since grid is a dynamic environment, solutions are always changing; therefore,, in this method, there is no need to keep information and previous history like classic glow worm . In addition, no massive particles are required. In this research, parameters of $\alpha$ , $\beta$ have been considered 1, and the value of $\gamma$ is from 0 to 1.

```
Particle Algorithm()
sourceLoad
while running
Do if job Queue.size>0
Then Lighteload        %% choose best
Neighbor(entekhab avalin behtarin)
SecondLightestLoad   %% choose SecondLightest
Neighbor(en entekhab dovomin behtarin)
TC (kamtarin hazine tebghe kamtarin faseleh)
threshold(tebghe avalin va dovomin)
while tc>threshold
do
Submit jobs        %%(TC)
sourceLoad         %%(currentNodeload)
velocity           %%(sourceload- lightestLoad)
```

***Figure 1: Running Algorithm by using glow worm***

### *3.1 Simulation Results*

After the proposed algorithm simulation in MATLAB environment, the results are compared with massive particles and genetic algorithm. The procedure of creating the network topology for the simulation is as follows:

At first, a minimum covering tree is structured, and then topology is obtained by adding the edges. In Figurer (2), time of the first job sending to the network and the last job are compared. The diagram showed the application of glowworm algorithm is more optimized than two other algorithms. In Figurer (3), the difference between the lightest and heaviest nodes have been indicated in terms of the load (the load in glow worm shows its own light, and due to its similarity with massive particles, we have shown job comparison with this algorithm.). In Figurer (4), the effects of increasing number of jobs on time have been shown. This time involves sending time of the first job and running the last job in the network.

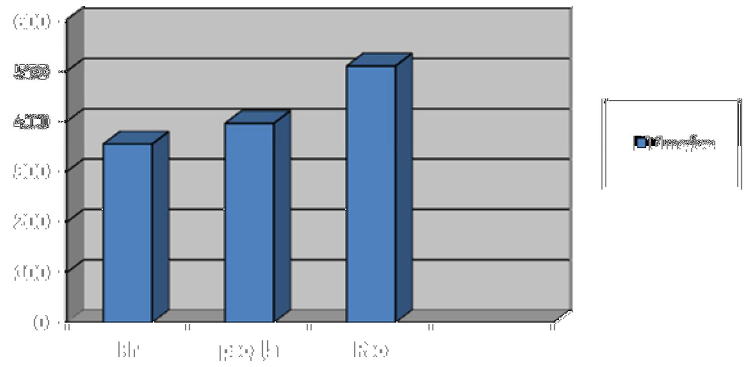

***Figure 2: the average of running time in different algorithms***

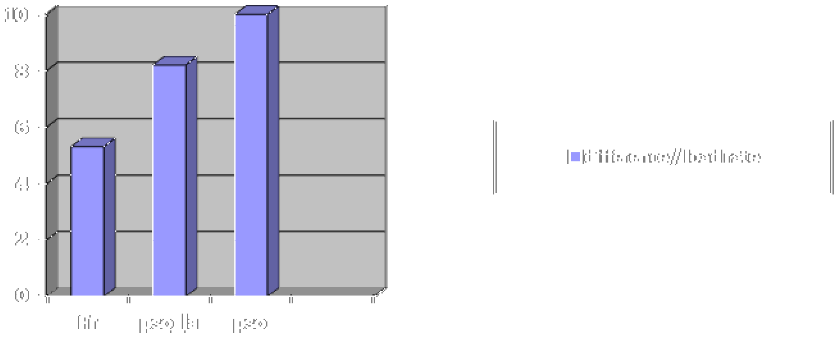

*Figure 3: the difference between load rate of the worm with less light and the worm with more light*

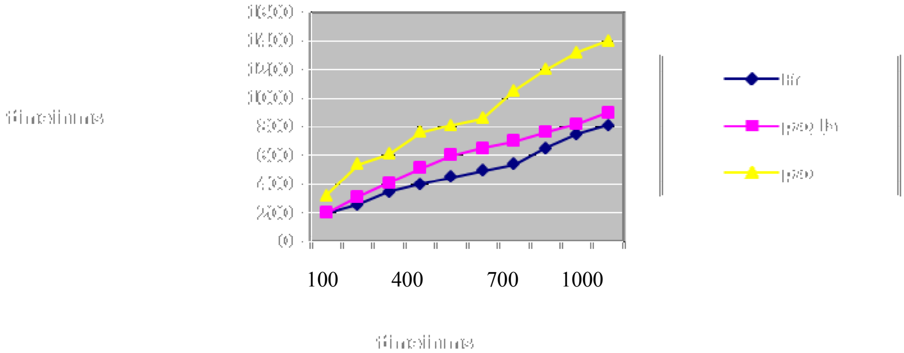

*Figure 4: increasing the number of jobs along with increasing the time*

## 4. Conclusion

In this paper, glow-worm algorithm was used to balance the load in Grid. The result showed that when there were various resources, this algorithm had more efficiency than other algorithms in terms of scheduling the functions and jobs. Glow-worm algorithm demonstrated better results in terms of running time. In addition, glow-worm algorithm was considered as one of the most suitable methods for optimization of the issues and problems.

## 5. References

[1] A. Al-Dahoud and M. Belal. "Multiple ant colonies for_ load balancing in distributed systems". In Proceedings_ of The _rst International Conference on ICT andAccessibility, 2007.

[2] C. Grosan, A. Abraham, and B. Helvik."Multiobjective evolutionary algorithms for scheduling jobs on computational grids".ADIS InternationalConference, Applied Computing( 2007), Salamanca,Spain, Nuno Guimares and Pedro Isaias (Eds.),( 2007).

[3] J. Cao, D. P. Spooner, S. A. Jarvis, and G. R. Nudd."Grid load balancing using intelligent agents". Future_Gener. Comput. Syst.,( 2005).

[4] J. L. Deneubourg, S. Aron, S. Goss, , and J. M._Pasteels. "The selforganizing exploratory pattern of_ the argentine ant". Journal of Insect Behavior, 1990.

[5] R. Subrata, A. Y. Zomaya, and B. Landfeldt. "Arti_cial_ life techniques for load balancing in computational_grids." J. Comput. Syst.Sci., ( 2007).

[6] J. K. J and R. C. Eberhart. Swarm Intelligence._Morgan Kaufmann Publishers, (2001).

[7] J. K. J and R. C. Eberhart. Swarm Intelligence._Morgan Kaufmann Publishers, 2001.

[8] Hassanzadeh, T. Meibodi, M, 1390. Presenting Improved Glowworm Algorithm for Optimization in Static Environment. The Fifth Conference of Datamining in Iran. Amirkabir University, Iran:Tehran.(1390).

[9] Y.S. Dai, M. Xie, K.L. Poh.," Reliability Analysis of Grid Computing Systems", Department of Industrial and System Engineering, National University of Singapore( 2010 ).