

# A Replica Selection Strategy On Ant-algorithm in Data-intensive Applications

<http://dx.doi.org/10.3991/ijoe.v9iS6.2799>

Ling YANG, Jinxian LIN and Yuying Zheng  
Fuzhou University, Fuzhou, China

**Abstract**—In data-intensive applications, multiple copies of the same data were created to improve reliability and reduce the bandwidth consumption. Replica selection is the crucial factor in accessing data quickly and effectively. It affects task scheduling, thereby affecting the efficiency and service quality of the application. Within this paper, a modified replica selection strategy based on ant-algorithm in data-intensive grid is proposed. The strategy is implemented and compared with other file replica selection algorithms on simulator OptorSim. The experiments results show that the strategy discussed in this thesis has certain advantages in balancing the load of sites and reducing the mean job time.

**Index Terms**—Data-intensive applications, Data grid, Ant-algorithm, Replica selection

## I. INTRODUCTION

As the rapid development of Internet and the popularity of networks, data plays a more important role in our life. Internet service quality is largely dependent on the data processing capability, and the management of large number of data causing performance bottlenecks in computers. In such a context, data-intensive computing (DIC) emerges and causes wide concern. Data grid, as the platform of data-intensive computing [1], has been concerned. Data grid technology allows geographically dispersed scientists to access and share the physically distributed resources, such as the computing resources, networks, storage resources and data collection which is the most important in large-scale data-intensive problems. The network latency and bandwidth between storage site and computing site affect a lot to the data grid performance. As a core service of data grid, the replica management has always been the core concerns of domestic and foreign researchers. Using file replica is the most common solutions in better file access time and bandwidth consumption, it is also an effective technique in improving the quality of service widely used in data-intensive computing. Replica strategy improves the system's performance and the robustness of distributed applications [2]. Storing multiple copies of the same data in different nodes, not only can improve the reliability and usability of data, but also can reduce data access latency, distribute storage site load. In many cases, replica management directly affect the efficiency and service quality of data-intensive applications. In general, replica management includes replica creation, replica selection, replica replacement, replica consistency.

In data-intensive applications, to improve the reliability and reduce the bandwidth consumption, multiple copies of the same data were created. In this case, many data needed by user tasks were stored in decentralized network sites. To access data quickly and effectively, there need to op-

imize replica selection to allow a job to choose one best copy, form those in a replica catalog, based on its performance and data access features. The selection of data replica affects task scheduling, thereby affecting the efficiency and service quality of the application. Therefore, replica selection is the crucial factor in data-intensive applications.

Replica selection algorithm has a number of static and dynamic metrics. Static metrics includes geographical distance and topological distance. Dynamic metrics includes run time, available bandwidth, file access time, characteristics of transfer, network status, replica host load and disk I/O information.

Two main solutions for replica selection were proposed. One is based on economic theory, such as game theory, auction model. Economic model can obtain a good overall performance. In [3], the author select replica base on KNN; in [4], the author proposed a strategy based on public auction protocol. The other strategy, based on swarm intelligence algorithm, makes up for the lack of economic model in finding the optimal copy in a single time effectively. In [5], Amin Shojaatmand proposed a dynamic ant-algorithm to select replica in data-intensive environment. This method improves the average access time to some extent. But the absolute choice, according to the value of a replica's pheromone, ignored the infinitely positive feedback characteristics of ant-algorithm. It will cause unlimited increase of replica host load. So there are some limitations. In [6], the author presented a combined algorithm based on ant-algorithm and genetic algorithm. It solves not only the inefficient disadvantage of genetic algorithm, but also the problem of solving problem slowly in the early period in ant-algorithm.

Within this paper, a modified replica selection strategy based on ant-algorithm in data-intensive grid is described. The grid simulator OptorSim [7,8] is extended and the strategy is implemented and compared with other algorithms on OptorSim. The experiment results show that the strategy proposed in this thesis has certain advantages in balancing the load of sites and reducing the mean job time.

## II. ANT-ALGORITHM

Ant-algorithm [9] was firstly proposed by M. Dorigo in 1991, it is based on ants foraging behavior of the real world. Same as genetic algorithm, it belongs to the heuristic algorithms, that is individual adjusts themselves and gradually converge to the optimal solution through dynamic interaction with external feedback. In the real world, ants will release pheromone to mark the path through when they forages, other foraging ants choose the closer path to the food according to the pheromone con-

centration, and leave their own pheromones. Pheromone on the closer path enhanced, so, all ants will choose the closer path to get food. Positive feedback and parallel characteristic of ant-algorithm make it suitable for solving the existing problems in distributed systems. The strategy mimics the behavior of how real ants find the shortest path from their nest to a food source.

### III. REPLICAS SELECTION ON ANT-ALGORITHM

Ants choose the closer path according to pheromone concentration, this paper applies such a theory to replica selection problem, and builds a model. The task node is visualized as foraging ants, the data file required by task is food, the replicas distributed in different sites are different paths to the food. Each replica has a eigenvalue, the size of eigenvalue indicates the possibility of being selected. The eigenvalue of replica changes with the different circumstances.

OptorSim uses replica locator, that is replica catalog, to organize and manage all the copies in network, replica manager to operate replicas, replica optimizer to control the specific way to select the replica. When a task wants to access a data file, the request firstly comes to replica manager. Then the replica optimizer searches the replica catalog according to a replica optimization algorithm, and chooses the best replica back to the manager. Finally, the best replica is delivered to user.

This article combines the idea of how ants find the shortest path and how a replica manager select the best replica. Some of Optorsim's modules are extended to investigate the new ant optimizer in, then simulate the optimized strategy. The new method assumes that each replica in the grid has a eigenvalue, that is pheromone. The initial value is calculated by its file size and I/O information. Generally, the larger value indicates the better replica. Pheromone is not a fixed value, it changes under different circumstances. When a replica is chosen and transmitted, the host load increases. To balance load, the pheromone should be decreased. If the remote access to a replica runs successfully, that the copy is available, the value should be increased; if it fails, the chance to be selected should be reduced, so the value decreases. When a request comes, replica optimizer inquiries replica catalog, calculates the selection probability of each copy, then selects the best one.

The metrics should be considered includes I/O read speed, bandwidth between request node and response node, replica host load [10].

The equations for the algorithm is as follows:

#### A. Initialization

When a replica  $i$  is created in the grid, the pheromone will be initialized based on (1).

$$T_i = r/f \quad (1)$$

Where  $T_i$  represents the pheromone,  $r$  denotes the read speed, it mimics the disk read speed of the SE in replica host. In OptorSim, there is no parameter  $r$ , so the SE is extended. The parameter  $f$  is the file size of the replica., and

#### B. Pheromone Update

When a replica is selected and transmitted, its pheromone changes. The pheromone is updated by

$$T_i' = T_i - k \quad (2)$$

$T_i'$  is the new pheromone value.  $k = f/b$ ,  $b$  denotes the available bandwidth between request site and replica host. There are several replicas in a SE, if some of them are selected, the SE load increases. To balance load, the pheromone of replica selected should be reduced.

When remote access returns, pheromone changes with two different circumstances. The corresponding equation is as

$$T_i' = (1 - v)(T_i + k) + ck \quad (3)$$

$v$  represents volatile coefficient, it is supposed to be 0.2.  $ck$  is the variance. When the transfer finishes, replica host load decreases, so a  $k$  value should be added. If the access to remote replica runs successfully, indicating that this replica is available. Variance  $ck$  is positive, here  $c$  is incentive factor supposed to be 0.8; if the access runs unsuccessfully, indicating that the replica is unstable. Variance  $ck$  is negative, here  $c$  is punish factor supposed to be  $-1.1$ .

#### C. Probability equation

$$p_i = (T_i)^\alpha \cdot (bnd_i)^\beta / \left( \sum_{u=1}^n (T_u)^\alpha \cdot (bnd_u)^\beta \right) \quad (4)$$

Where  $i, u$  belong to Replica sets of the same data. Assume that there are  $n$  replicas for a data file.  $T_i$  is the pheromone,  $bnd_i$  is the available bandwidth between two communication nodes. To avoid the infinite positive feedback of ant-algorithm causing unlimited increase of replica host load. The new algorithm selects the best replica according to probability distribution: assume that there are  $n$  replicas, and calculate each replica's probability  $p_i$  ( $1 \leq i \leq n$ ) based on (4), let  $p_0 = 0$ ,  $ps_i = p_0 + p_1 + \dots + p_i$  ( $1 \leq i \leq n$ ), use random function generate a random number  $rdm$  ( $rdm \in [0, 1]$ ). If  $ps_{i-1} \leq rdm \leq ps_i$ , then the replica  $i$  is chosen as the best replica. The probability of  $rdm$  be in the area between  $ps_{i-1}$  and  $ps_i$  is equal to  $p_i$ . Thus, this method make sure the replica with the most pheromone has max probability to be chosen in a long term. Flowchart of algorithm is shown as Fig.1.

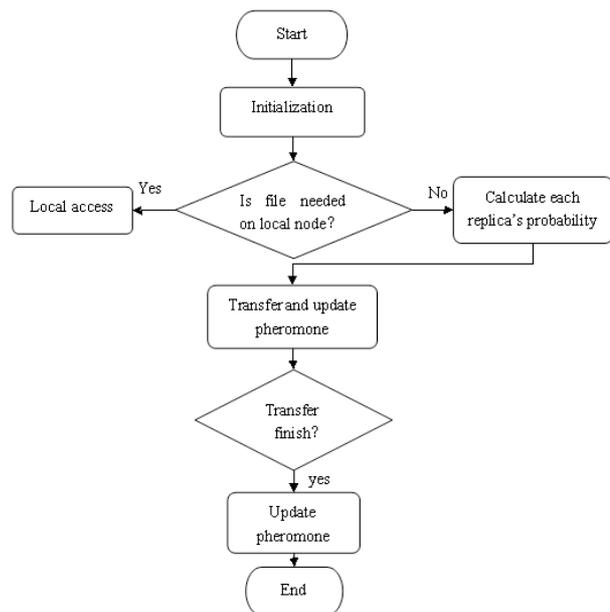


Figure 1. Flowchart of algorithm

IV. SIMULATION

A. Simulation Configuration

The experiment program is based on European data grid test bed (edg\_testbed). It includes 18 sites distributed in different countries and regions. Each site has 0 or 1 CE, and 0 or 1 SE. Those have no SE and CE are seen as routers. The topology of edg\_testbed is shown in Fig.2.

Hardware conditions of simulation: CPU 1.87GHz, Memory 1G.

Job configuration:six kinds of jobs are in Table 1. Each kind needs different data files, and has different probability of being selected. Site with a CE can run all the six kinds of jobs. The number of jobs could be set in configuration table.

Initial distribution of data files: the size of SE in site 8 (CERN) is 100GB, it's big enough to storage all the source files. So all the files needed by jobs can be find here.

B. Simulation Results and Discussion

This thesis aims to discuss the replica selection problem, so creation and replacement problem will not be discussed. To evaluate the efficiency, the algorithm is compared with other two methods. One is described in [7], the other is already in OptorSim called SimpleOptimiser. Both never replicate a file.

The criteria for replica selection includes Mean Job Time, Effective Network Usage and SE Usage. Running the same set of experiments with same configuration, so the different algorithm have the same SE Usage. There is no difference in ENU based on (5), so just compare their MJT here.

$$ENU = (N_{rfa} + N_{fr}) / (N_{rfa} + N_{lfa}) \quad (5)$$

where  $N_{rfa}$  denotes the number of remote file access,  $N_{fr}$  is the number of file replicas,  $N_{lfa}$  represents the number of local file access.

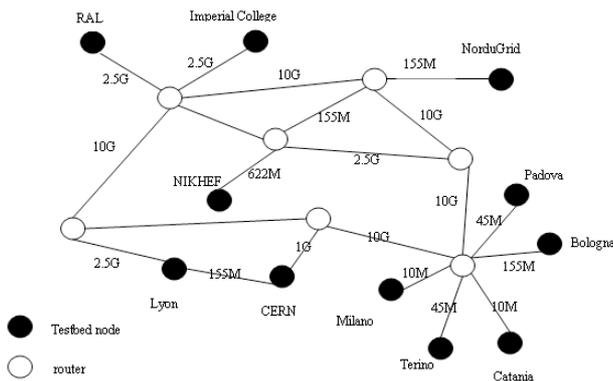


Figure 2. Network topology of edg\_testbed

TABLE I. JOB CONFIGURATION

Job Name	Data Files	Probability
jpsijob	12	0.17
highptlejob	2	0.17
incelejob	5	0.17
incmuonjob	14	0.16
highptphotjob	58	0.17
zbbbarjob	6	0.16

The file's access pattern is set to be sequential, scheduling algorithm of RB is access cost for current job and all queued jobs. Compares among the three methods under different number of jobs were done. The experiment parameter configuration is shown as Table 2.

To evaluate the new algorithm more efficiently, an experiment is repeated 5 times and its average value is calculated. The simulation results are shown in Fig.3.

The evaluation demonstrates the new algorithm reduces Mean Job Time of the grid system. Compared to SimpleOptimiser, the method gets better performance with different number of jobs; compared to OldAntOptimiser [7], the new method select best replica according to the probability distribution, but not the one with the biggest pheromone. Thus, it makes sure the replica with the most pheromone has max probability to be chosen in a long term, and avoids the infinite positive feedback of ant-algorithm causing unlimited increase of replica host load. So the strategy will get better performance with large number of jobs.

V. CONCLUSIONS

As a core service of data grid, the replica management is an effective technique, widely used in data-intensive computing, in improving the service quality. The new strategy in this paper improves the OldAntOptimiser, propose a new strategy for replica selection. Simulation result with the optorsim demonstrates that the strategy could improve mean job time and balance network load to some extent. At present, the new algorithm runs in circumscribed simulation conditions. Expecting it to be used in real network in the future, many revise should be done, and the advantages is yet to be excavated.

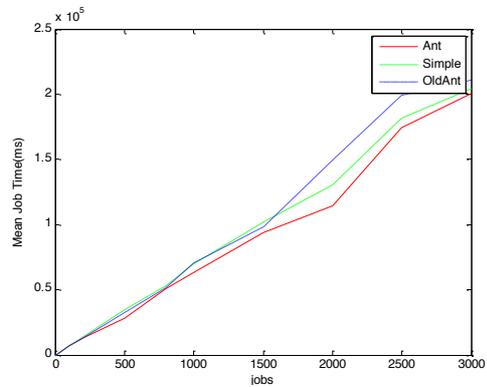


Figure 3. Compare of three algorithms

TABLE II. PARAMETER CONFIGURATION

Optimisers	Ant/Simple/OldAnt
Number of jobs	100/200/500/800/1000/ 1500/2000/2500/3000
Grid Configuration file	edg_testbed_grid.conf
Job Configuration file	edg_testbed_jobs.conf
Scheduling Algorithms	access cost for current job + all queued jobs
Access Pattern	SequentialAccessGenerator
Job Delay	25000ms
Max Queue size	200
Users	Simple
Initial File Distribution	8
Fill All Sites	yes

## REFERENCES

- [1] F.Y Shi, "Research on Replica Optimization in Data-intensive Computing", *M.A.Thesis*, Wuhan University of Technology, Wuhan, 2011.
- [2] A.Moallem and S.A.Ludwig, "Using Artificial Life Techniques for Distributed Grid Job Scheduling", in *Proceedings of the 2009 ACM symposium on Applied Computing*, Honolulu, Hawaii, 2006.
- [3] R.M.Rahman, R.Alhajj, and K.Barker, "Replica selection strategies in data grid", *Journal of Parallel and Distributed Computing*, Vol.68, pp.1561-1574, 2008. <http://dx.doi.org/10.1016/j.jpdc.2008.07.013>
- [4] X.D You, "Research on Replica Management in Data Grid", *M.A.Thesis*, Northeastern University, Shenyang, 2005.
- [5] A.Shojaatmand, N.Saghiri and S.Hashemi, "Improving Replica Selection in Data Grid using a Dynamic Ant Algorithm", *International Journal of Information Studies*, Vol.3, pp.139-144, 2011,
- [6] Z.W Guo, N Li, H Li and N Shi, "New Replica Selection Strategy Based on Combination Algorithm", in *International Conference on Internet Technology and Applications*, Wuhan, China, 2010. <http://dx.doi.org/10.1109/ITAPP.2010.5566297>
- [7] L.J Zhou, "Analysis and Improvement of Grid Simulator Oport-Sim", *M.A.Thesis*, Tianjin University, Tianjin, 2006.
- [8] W.H.Bell, D.G.Cameron, L.Capozza, A.P.Millar, K.Stockinger, "Oportsim-A Grid Simulator For Studying Dynamic Data Replication Strategies", in *Centro per la Ricerca Scientifica e Tecnologica (ITC-IRST)*, Trento, Italy, June 2003.
- [9] H.B Duan, D.B Wang, J.Q Zhu and X.H Huang, "The Improvement of Ant Colony Algorithm Theory and application research", *Control and Decision*, Issue 12, 2004, pp.1321-1326+1340.
- [10] M Sun, J.Z Sun and E.F Lu, "Ant Algorithm for File Replica Selection in Data Grid", in *Proceedings of the First International Conference on Semantics, Knowledge, and Grid(SKG 2005)*, Tianjin, China, 2005. <http://dx.doi.org/10.1109/SKG.2005.43>

## AUTHORS

**Jinxian LIN** is with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, CO 350108 China (e-mail: jxlin@fzu.edu.cn)

**Ling YANG** is with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, CO 350108 China (e-mail: 750362367@qq.com)

**Yuying Zheng** is with the College of Material Science and Engineering, Fuzhou University, Fuzhou, CO 350108 China (e-mail: yyzheng@fzu.edu.cn)

This article is an extended and modified version of a paper presented at the 2012 International Conference on Artificial Intelligence and Its Application in Industry Production (AIAIP 2012), held in Wuhan, China in December 2012. Manuscript received 19 May 2013. Published as resubmitted by the authors 26 June 2013.