A Trust Based Ant Colony Optimized Grid Scheduling

E. Saravana Kumar [1], A. Sumathi [2]

[1.] Research Scholar, Anna University of Technology, Coimbatore, Tamil Nadu, India
[2.] Professor/ECE, Adhiyamman College of Engineering, Hosur, Tamil Nadu, India
sarankr.info@gmail.com

Abstract: Resources discovery and scheduling, a core grid service impacts overall system performance and service quality directly. In grid computing services, the system in addition to attempting to locate an optimal resource to improve overall system performance also aims to use resources efficiently. Most scheduling and resource selection algorithms fail to consider resource characteristics resulting in reliability and security issues which in turn affect service quality. This work proposes implementation of a Trust Ant colony optimization (TACO) module, forecasting trust throughout a network grid. The proposed approach computes each entity's Trust Factor value to determine the self-protection capability and reputation weightage. When the trust module is integrated with dynamic scheduling, lower task failure ensures secure resource utilization.
[E. Saravana Kumar, A. Sumathi. **A Trust Based Ant Colony Optimized Grid Scheduling**. *Life Sci J* 2013;10(7s): 466-470] (ISSN:1097-8135). http://www.lifesciencesite.com.

Keywords: Grid Computing; Resource Selection; Trust, Reputation; Ant Colony Optimization

1. Introduction

The Grid approach can access, utilize, and manage various heterogeneous resources across multiple domains in virtual organizations and institutions. Grid computing [1] first emphasised large-scale resource sharing, innovative applications, and high-performance achievement. Currently, a Grid approach [2] suggests a distributed service environment development integrating varied resources with quality of service capabilities to support scientific/business problem solving environments. But, distributed services/resources optimal utilization usually needs a Grid user to decide the capacity of remote resources. Users want to identify possible candidate resources through meta information from directories, databases/registries in a grid scenario.

But, the current generation of Grid information services provides only elementary information to guide sophisticated quality of service based resource selection process. The Globus Monitoring and Directory Service (MDS) [3] provide limited information about Grid resources including static and possibly dynamic properties. In many cases, information returned by the service is costly, inaccurate/out-dated without integrating a resource selection service. Selecting appropriate resource within a Grid environment which satisfies QoS (quality of service) requirements is challenging. Without a high degree of confidence relationship, one cannot attain efficient resource allocation and use.

Security is important in a grid environment involving every grid computing stage including resource selection, job submission, secure communication, authentication and authorization [4]. These can be called attribute based security mechanisms. Literature has many mechanisms for attribute based grid security [5, 6, 7]. Trustworthiness identification of resource, based on earlier behaviour and grid interaction is an emerging research area.

Understanding that some individuals could be dishonest is the fundamental motivation for work on Trust and Reputation Systems (TRSs). A new network model was established for reputation computation in virtual organizations, and its dynamics were investigated by Jinde Cao et al [8]. Many conditions ensured global asymptotical stability of the new network model using Lyapunov method and Linear Matrix Inequality (LMI) technique where stability ensures that entities reputation degrees is able to tend some constants with time.

Centrally managed traditional security systems limit collaborative action among entities in open networks (like Grids). This needs new methods to handle security in large distributed systems and new research specially in areas concerned with security provision through collaboration. Hassan et al [9] presented a design of large-scale, self-managing Trust Management Framework (TMF) which efficiently uses invisible evidence scattered across global networks. TMF design includes a layered architecture to capture evidence at network data layer, transforming it into formed reputations in information layer, using such reputations to determine entity trustworthiness in a network's knowledge layer. TMF automated scattered evidence acquisition and formulation, evolution and dissemination of reputations in a scalable way to ensure improved security decisions.

This work proposes implementation of a trust Ant colony optimization (TACO) module, predicting trust across the grid network. When trust module and dynamic scheduling are integrated, task failure decreases by securing utilized resources.

2. Related Works

The current work is inspired by many earlier works relating to trust management and reputation based security enhancement to sustain grid computing performance. Some related works are reviewed below.

Sonnek, et al., [10] selected many reputation algorithms adapting them to service selection problems in a Grid-like environment. Quantitative comparisons of accuracy and overhead associated with such techniques under common scenarios were undertaken with the results proving that a reputation system to guide service selection greatly improved client satisfaction with minimal overhead. It was also demonstrated additionally, that the appropriate algorithm depended on anticipated attacks. A reputation system using global ratings worked better if clients were honest, while a system using local ratings was better at thwarting misbehaving clients. Results demonstrated that a guide service selection reputation system could improve client satisfaction greatly with minimal overhead. Results also show a significant difference in different ranking algorithms performance depending on system properties.

Alunkal, et al., [11] suggested a reputation management framework for Grids to ensure a distributed/efficient mechanism for resource selection. The proposed reputation management service is dependent on dynamic trust concept, and reputation adaptation based on community experiences to classify, select, and tune entities allocation, including resources, service, and people provided services. The framework evaluates simple contextual quality statements through specialized services to effect a monitored resource's reputation. The proposed reputation service uses a novel algorithm to evaluate Grid reputation through a combination of two known concepts including (a) using eigenvectors to compute reputation and (b) integrating global trust. The resulting framework was called GridEigenTrust framework.

Bawa, et al., [12] proposed an approach to provide reliability and reputation aware security for resource selection in a grid environment. In the proposed approach, self-protection capability and reputation weightage obtain Reliability Factor (RF) value. Hence jobs are allocated to resources with higher RF values. The proposed approach aggregates several security related attributes for self-protection capability and numerical values reputation, which can calculate a grid entity's Trust factor. Experimental evaluation revealed that when higher trust/reliable nodes are chosen failure chances decrease greatly.

Gupta, et al., [13] proposed a multi agent grid architecture where grid entities (users/resource providers) have their own agents. Users identify reputed resource providers through the application of Fuzzy inference system recommended by trustworthy peers. When a user agent gets recommendations from trustworthy acquaintances, they aggregate recommendations to identify trustworthy resource providers. The user agent then forwards a resources access request to reputed resource providers, who apply binary logistic regression on factors influencing a decision on allowing a requesting user to use the grid. Simulations demonstrate the high probability resource provider can predict a malicious user as not trustworthy.

Kavitha, et al., [14] new method provides past interactions and present environment characteristics based quantitative trust value which selects a suitable job resource and eliminates run time failures due to incompatible user-resource pairs. The proposed work can calculate grid components trust values improving success rates of jobs submitted to grid resources. Resource access depends not only on resource identity and behaviour but also on transaction context and time, connectivity bandwidth, resource availability and resource load. The recommender quality, based on resource feedback accuracy is also evaluated. Jobs are submitted for execution to a selected resource after discovering a resource's overall trust value which is then computed with regard to subjective/objective parameters. Job failure rates are greatly lowered, according to results.

3. Methodology

3.1 Trust

Trust and reputation vectors combine to compute a novel Trust Factor Computation (T_a) for resource selection. The proposed Trust Factor is measured using resources trust with resource reputation. Trust computation's key parameters are as follows:

Intrusion Detection System Capabilities (IDSC): An entity's ability to safeguard a system against host/network based intrusions.

Anti-virus capabilities (AVC): An entity's capability to secure a system against viruses and malicious codes.

Firewall capabilities (FC): Capability to guard the entity against other network accesses.

Authentication Mechanism (AM): A mechanism's capability to authenticate an identity claimed by/for system security.

Secured File Storage Capabilities (SFSC): An entity's ability to store files needed for job execution securely.

Interoperability (I): An entity's ability to confine interfacing amid concurrent jobs.

Secured Job Execution (SJE): An entity's capacity to ensure a secure job execution.

Authorization mechanism (A): Mechanism to determine access level with a specific authenticated user.

Self-Protection Capability (SPC): A security factors value aggregation given in definitions 1 to 8 to determine and entity's Self Protection Capability (SPC).

Weightage is allocated to security factors, based on security and aggregated at the final point to calculate self-protection capability, calculated as follows using equation,

$$SPC = \sum_{i=1}^{n} W(i) * A(i)$$

where n is the total number of factors,

W is the weightage

$A(i)$ is the value of the factor.

An entity's Reputation (R) is determined on feedback provided by the user community and other grid entities, as regards the entity's security characteristics and earlier experiences. When completed feedback is given by the user on Reputation Manager (RM) attributes depending on their experience. Similarly, grid entities offer timely feedback to the RM. Feedback's value is in the 0 – 1 range. Users and entities feedback are then aggregated.

An entity's Trust Factor (T) is determined through its Self-Protection Capability (SPC), and Reputation computed through the following equation:

$$TF(E_a) = SPC(E_a) + RW(E_a)$$

3.2 Resource Selection

Grid scheduling is a very difficult combinatorial problems in classical scheduling theory and is NP-hard (nondeterministic polynomial time-hard).The grid system's heterogeneous, dynamic, and distributed nature makes it hard to capture a physical network topology graph. Information is hard to get from routers, switches and other network devices configured/managed by different administrative domains due to reasons of technology, administrative and security reasons. But complete knowledge of the physical network topology is not required to solve basic resource selection issues. Normally, networks coarse and logical description connecting computing nodes is enough. In this article, grouping "nearby" nodes into clusters and computing the "distance" between them is implemented. This information is enough for a scheduler to determine which node groups can be used to solve smaller problems and which cluster combination are competent for larger problems.

The K nearest algorithm computes the relationship between two clusters using the

$$K_d = \{A,F,Au,B,C,T,R,S\}$$

where

K_d is the distance metric

A is the antivirus capability

F is the firewall capability

Au is the authentication mechanism

B is available bandwidth

C is the available computing cycle

T is the trust

R is the reputation

Distance between two encoder values gives the relationship between two resources to match similarity.

$$d_m = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

$$\text{with } d_{m(i)} = \begin{cases} 0 \ when \ r_{m(i)} \le 0 \\ 1 \ when \ r_{m(i)} \ge 1 \end{cases}$$

x_i is the encoder value

Values equal to 1 show strong affinity and near 0 have weak affinity for cluster formation. When a cluster is formed Ant Colony Optimization (ACO) is used for makespan computation.

3.3 Proposed Ant Colony Optimization

ACO algorithms were inspired by observing real life ant colonies [15, 16]. An interesting behaviour is how they locate the shortest routes between food sources and the nest. When going from a food source to the nest and vice-versa, ants deposit a substance named pheromone which they can smell. Thus, they choose paths marked by strong pheromone concentrations. The pheromone trail allows ants to find the way both to the nest and food source.

The ACO algorithm uses a colony of artificial ants behaving as co-operative agents in a mathematical space where they search and reinforce pathways (solutions) to locate optimal ones. A solution that satisfies the constraints is feasible. After pheromone trail initialization ants construct feasible solutions, starting from random nodes. Then

pheromone trails are updated. Figure 1 shows the ACO algorithm's flow.

```
Procedure ACO
begin
    Initialize parameters, the pheromone trails
    while (stopping criterion not satisfied) do
        Each ant position starts at starting machine
        while (stopping when every ant has built a solution) do
            for each ant do
                Choose next machine by applying the state transition rule
            end for
        end while
        Update the pheromone
    end while
end
```

Figure 1: Ant Colony Optimization

Grid computing is a heterogeneous and dynamic environment. Scheduled jobs rarely coincide with the actual execution times and expected ones in a real computing environment. Hence, the scheduling job's challenge is in the grid as no one has the ability to control all jobs fully. The other challenge is in dynamic resources availability and difference between the expected execution time with actual time in algorithm.

The generated concrete workflow's reliability must be not smaller than a user-defined variable Reliability Constraint. Reliability constraint is based on Trust and Reputation. Total workflow execution time should not be larger than a user-defined variable Deadline. In other words, if E is the total execution time of N jobs, then E should satisfy E = Deadline.

Pheromone and heuristic information are important ACO algorithm factors. Generally, pheromone records historical searching experiences and bias ants' future search behaviour. On the other hand, heuristic information has problem-based values to guide ants search direction. As a scheduling problem aims to map tasks in an abstract workflow to service instances to form a concrete workflow, the pheromone value of mapping service instance s_{ji} to task T_i as t_{ij} (phermone value) , and the heuristic information value of mapping s_{ji} to T_i as n_{ij}.

$$\eta_{ij} = \frac{current\_trust - \min\_trust}{(Max\_trust - avg\_least\_5\_trust\_values)}$$

$$t_0 = \frac{\min\_reliability}{\max\_reliability}$$

Local Pheromone is updated immediately after ant maps a task T_i to a service instance s_{ji}. Local pheromone updating rule is applied to reduce the attraction of s_{ji} for the later ants. The local pheromone updating rule is given by the following equation:

$$t_{ij}^{new} = (1-\rho)t_{ij} + \rho t_0$$

where ρ is a value between 0 and 1. in this study, ρ is assigned 0.5.

0.4(min_reliability/max_reliability)+0.6(min_makespan/max_makespan) is the criteria that have to be obtained for selecting the ideal node.

4. Results and Discussion

Experiments were simulated in a grid environment with five clusters and 100 jobs. The make span is defined as the maximum time required to complete all meta-task jobs. Makespan is applicable for batch jobs and measures scheduling performance. Figure 2 shows the makespan achieved for different generations.

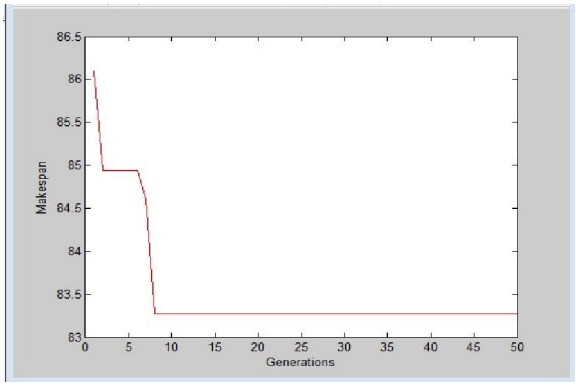

Figure 2: Makespan Achieved.

From figure 2, it is observed that for 5 clusters with 100 jobs the average makespan obtained is 83.2

5. Conclusion

The proposed scheduling algorithm is designed to achieve high throughput computing in a grid environment. This is a NP-problem requiring exponential time for a solution. Hence a heuristic algorithm is developed to find a good solution in a reasonable time. This paper discusses a heuristic algorithm based on ACO method and basic strategies for grid scheduling are formulated. This work proposes implementation of a Trust Ant colony optimization (TACO) module, predicting trust across a grid network. The approach computes each entity's Trust Factor value to determine the self-protection capability and reputation weightage. Task failure decreases when integrating trust module with dynamic scheduling through securing used resources.

References

1. Foster and C. Kesselman, Eds., The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, July 1998.
2. G. von Laszewski, G. Pieper, and P. Wagstrom, "Gestalt of the Grid," in Performance Evaluation and Characterization of Parallel and Distributed Computing Tools, ser. Series on Parallel and Distributed Computing. Wiley, 2003.
3. G. von Laszewski, S. Fitzgerald, I. Foster, C. Kesselman, W. Smith, and S. Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations," in Proceedings of the 6th IEEE Symposium on High-Performance
4. Erik Elmroth. and Johan Tordsson. "Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions", Future Generation Computer Systems, Vol. 24, pp. 585-593, 2008.
5. Humphrey. M, Sang-Min Park, Jun Feng, Beekwilder. N, Wasson. G, Hogg. J, LaMacchia. B.; Dillaway. B, (2007). Fine-grained access control for GridFTP using SecPAL. 8th IEEE/ACM International Conference on Grid Computing, pp.217-225.
6. Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R., & Freeman, T. (2006, July). A multipolicy authorization framework for grid security. In Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on (pp. 269-272). IEEE.
7. Metke, A. R., & Ekl, R. L. (2010). Security technology for smart grid networks. Smart Grid, IEEE Transactions on, 1(1), 99-107.
8. Cao, J., Yu, W., & Qu, Y. (2006). A new complex network model and convergence dynamics for reputation computation in virtual organizations. Physics Letters A, 356(6), 414-425.
9. Mohammad Waseem Hassan., Richard McClatchey. and Ian Willers. "A Scalable Evidence Based Self-Managing Framework for Trust Management", Information Sciences, Vol. 179, No. 15, pp. 2618-2628, 2007.
10. Sonnek, J. D., & Weissman, J. B. (2005, November). A quantitative comparison of reputation systems in the grid. In Grid Computing, 2005. The 6th IEEE/ACM International Workshop on (pp. 8-pp). IEEE.
11. Alunkal, B., Veljkovic, I., Von Laszewski, G., & Amin, K. (2003). Reputation-based grid resource selection. Proceedings of AGridM.
12. Bawa, R. K., & Sharma, G. (2012, June). Decision based Resource Selection in Grid Environment. In IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT 2012)(No. 5, pp. 16-19). Foundation of Computer Science (FCS).
13. Gupta, B., Kaur, H., Namita, N., & Bedi, P. (2011, June). Trust Based Access Control for Grid Resources. In Communication Systems and Network Technologies (CSNT), 2011 International Conference on (pp. 678-682). IEEE.
14. Kavitha, G., & Sankaranarayanan, V. (2010, December). Secure Resource Selection in Computational Grid Based on Quantitative Execution Trust. InProc. Of International Conference on Information, Communication, Electrical and Computer Engineering, Singapore (pp. 149-155).
15. Dorigo, M., Di Caro, G.: The Ant Colony Optimization metaheuristic. In: New Idea in Optimization, Corne, D., Dorigo, M., Glover, F. eds., McGrow-Hill (1999) 11–32.
16. Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transaction on Evolutionary Computation, 1 (1999) 53–66.

1/8/2013