Journal of Computer and Communications, 2015, 3, 34-41
Published Online June 2015 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2015.36005

Trust Management in Grid-Trust Assessment and Trust Degree Calculation of a Resource—A Novel Approach

Avula Anitha

School of Informatics, Institute of Technology (IOT), Hawassa University, Hawassa, Ethiopia
Email: gamyav@yahoo.in

Received 21 April 2015; accepted 25 May 2015; published 28 May 2015

Abstract

Grid Computing is concerned with the sharing and coordinated use of diverse resources in distributed Virtual Organizations. This introduces various challenging security issues. Among these trusting, the resources to be shared and coordinated with the dynamic and multi-institutional virtual organization environment becomes a challenging security issue. In this paper, an approach for trust assessment and trust degree calculation using subjective logic is suggested to allocate the Data Grid or Computational Grid user a reliable, trusted resource for maintaining the integrity of the data with fast response and accurate results. The suggested approach is explained using an example scenario and also from the simulation results. It is observed that there is an increase in the resource utilization of a trusted resource in contrast to the resource which is not trusted.

Keywords

Grid Computing, Virtual Organization, Computational Grid, Data Grid, Trust Management, Trust Assessment, Trust Degree Calculation

1. Introduction

The term "Grid" is frequently used to refer to systems and applications that are integrated and distributed across multiple control domains. A common scenario in Grid Computing is e-business scenario, *i.e.* the formation of dynamic virtual organizations "VOs" comprising groups of individuals and associated resources and services united by a common purpose but not located within a single administrative domain. The need to support the integration and management of resources within such VOs introduces challenging security issues [1]. The relationships among participants, *i.e.* clients in VOs represent an overlay with respect to the relationships existing

How to cite this paper: Anitha, A. (2015) Trust Management in Grid-Trust Assessment and Trust Degree Calculation of a Resource—A Novel Approach. *Journal of Computer and Communications*, **3**, 34-41.
http://dx.doi.org/10.4236/jcc.2015.36005

between those participants, the resources and owners of the resources. This overlay exists both in terms of trust and with respect to the security mechanisms and policies in place at those organizations, resources and their owners [2] as shown in **Figure 1**.

This paper addresses the trust management issues in grid computing for maintaining the integrity and non-repudiation. It gives a mechanism where the user can assess the trust on a resource and gives the feedback about the resource by considering the service it has received from the allocated resource. While giving the feedback various factors such as response time, delay time, accuracy of results and the integrity of the data for its job when it is executed on that resource are considered. Depending on the feedback the trust degree is calculated and maintained in the recommendation table and feedback table by the resource allocation manager. In addition, the simulation is done, in which the grid users and grid resources with various specifications are generated. In this simulation, the trust assessment about a resource is done by the users and the trust degree or level of trust is calculated by the resource allocation manager. Further, the analysis is done by using an example scenario by calculating the trust degree and this is compared with the scenario where a trust degree is not considered while allocating the resource. The rest of this paper is organized as follows: Section 2 of this paper discusses the related work. Section 3 discusses the motivation and approach for the proposed work. Section 4 gives the simulation for assessing and calculating the trust degree of a resource using an illustration and performance results are analyzed. Section 5 ends with the conclusion and future work.

2. Related Work

This section discusses about the related work. Grid computing research has produced security technologies based not on direct inter-organizations trust relationships, but rather on the use of VO as a bridge among the entities participating in a particular community or function [2]. The secure grid environment must employ mechanisms to secure authentication, authorization, data encryption, resource protection and secure communication. Grid security itself presents several unique security challenges, including managing user identities [3] across local and global networks, managing the diversity of local resources/user security systems trust relationships between entities, end-user key and credential management and providing security to resources against malicious acts of grid users [4]. Designing a secure grid implies taking into account the needs of grid users to secure remote resources that protect the integrity and confidentiality of data and also the needs of resource owners to ensure that only authorized, trustworthy individuals are using their systems [5]. Cody *et al.* have separated the security solutions into system based and behavior based solutions. For example, a grid developer/administrator looking to implement a specific technology as his grid system might turn immediately to the system-based solutions, while a researcher studying behavioral aspects of security of the classification system can use behavioral solutions [6]. Trust management will fall under the category of behavioral solutions. Trust differs from authentication. Authentication seeks to expose the identity of someone attempting to gain access to the grid. The

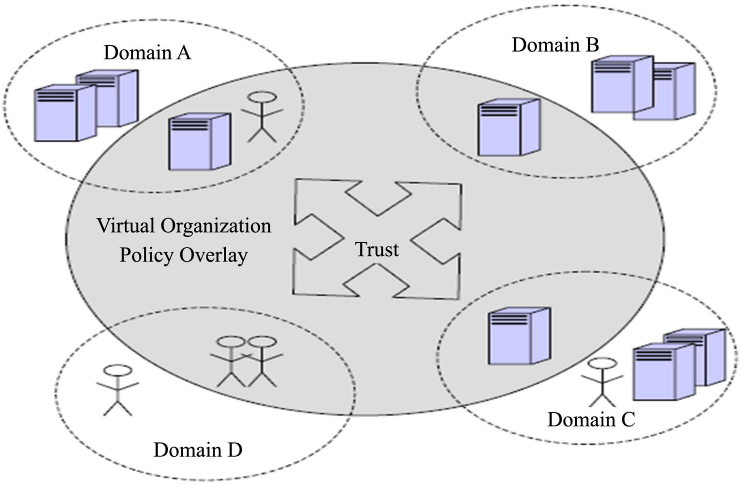

Figure 1. A virtual organization policy domain overlay pulls together participants from disparate domains into a common trust domain.

Trust-based solutions by contrast, go one step beyond, from providing an individual identity to associating a level of trustworthiness with that identity. A user of the system can make better decisions about the interaction with its peers if it knows the reputation of that peer in the system. Creating the notion of a global trust value for each user in the system can lead to segregation of the proper use of the system from the misbehaving users of the system. In the Eigen Trust algorithm [7], a resource sets its Required Trust Level (RTL); a client must have to access a resource using direct experience, reputation and time since the last interaction with the entity in question. In this a trust agent evaluates the level of trust, based on the direct trust relationship and on the reputation from recommender entities in the grid. The client also sets RTLs of resources they wish to use. If a client and resource have compatible trust levels the operation they are involved in goes on without additional security overheads. One of the problems of scheduling resources on a grid is that it is hard to know how long a resource will be available for or how good its performance will be if it is used [8].

Researchers have implemented a tool known as Every Ware [9], which contains amongst other things, a performance forecasting mechanism, which makes the scheduling simpler by knowing the reaction of resource to requests faster or data processing faster. The data related to resource performance can be collected using direct experience [7] or through Grid Monitoring systems [10] [11] which provide capture, analysis, logging and visualization of monitoring data aggregated from grid resources, such as web services (state information) and Fabric layer resources (e.g. Hardware characteristics) across a set of nodes available via the grid. Grid Monitors may themselves be present in data and/or compute grids—one such pure grid monitoring system is Ganglia. Srivaramangai *et al.* [12] proposed a trust model to improve reliability in the grid. According to their model, reputation based systems can be used in grid to improve the reliability of transactions. To achieve reliable transactions, mutual trust must be established between the initiator and the provider. Two types of trust have been taken, namely direct trust and indirect trust. Indirect trust is measured from the reputation score of other entities. Wang Meng *et al.* [13] proposed a Dynamic Grid Trust Model named Dy Grid Trust which is based on recommendation credibility. This model suggests a way to distinguish honest and dishonest recommendation and adjust the weight of trust evaluation dynamically. [14] propose a novel trust model reflecting the required dynamic nature of trust for grid entities, through cross organizational boundaries, with little administrative overhead. Based on cross domain grid computing GPC-AKA authentication protocol, a Grid Trust Management (GTM) model has been designed to establish trust relations between grid entities. [15] proposes a new method to the ability of probability theory for managing the behavior trust in grid computing systems.

3. Proposed Work

In this section, the motivation and the approach used in the proposed work are discussed.

3.1. Motivation

In Grid Computing, resource owners are often hesitant to enter the grid environment because they will be sharing viable resources [1]. This distrust leads many potential grid entrants to use their own "closed-box" system rather than a grid system with other resources. This is an inefficient use of global computing resources which can be addressed through the use of trust-based security solutions in grid computing. By trustworthiness these solutions are inferring the belief that a particular user will use the resource in a non-malicious manner. In the SETI@home project [16], the work of volunteers around the world, allowing their computers to be used for scientific research shows that some people, at least are willing to share for no direct benefit to them but it is unlikely that everyone would allow this. Within single businesses or university departments it is likely that it could be social policy that every computer must be part of the organization's Grid, but this would probably not work for the Grid without some sort of global billing system. The Java Sandbox Security Model [17] already provides an environment in which untrusted users are restricted from making certain system calls which are not considered safe, and from accessing memory addresses outside of a certain range. Any Grid system will have to provide a similar mechanism, so that users will be happy to let others access their computer. Trust based solutions can be identity based [3] or behavior based. Identity based solutions deal with who you are, while a behavioral based solutions deal with what you do. Trust based solutions function in computational and data grids. In Grid, certificates alone cannot validate authenticity. The trust in the binding of a certificate key and its owner is essential in providing a level of legal culpability (*i.e.* Digital certificates and non-repudiation). The certification authority that created the certificate must also be assessed for trustworthiness. Do they properly check identification before issuing a certificate? The authenticity of a key can be validated with its corresponding public or private key. However the certificate that holds the key is what needs to be validated. Rather, digital certificates will be used only for authenticating the users and resources. The digital certificates will not give the information about how reliable or trustworthy is a resource in terms of computing performance and also in terms of maintaining the data integrity. Therefore, the clients and resources should have the proper trust on each other rather than depending only on digital certificates.

3.2. Proposed Work—An Approach

In the proposed work, the trust on a resource is measured using subjective logic. It is in the form of degree of belief, disbelief and uncertain. If the trust degree is "belief" then the resource can be trusted completely and can be allocated to the user. If the trust degree is "unbelief" then the resource cannot be trusted and it should not be considered at the time of allocation and if it is "uncertain", then the resource should be allocated in case if no resource with a trust degree belief is present in the list of available resources.

In this, trust management is done in 2 phases. 1) Creating the trust relationship, a trust assessment is done using different parameters; 2) Calculation of Trust degree of a resource.

3.2.1. Trust Assessment

In this phase the participants or resource users can give their feedback after accessing the resource and after getting the outputs or results from the resource. The feedback can be given by taking into account the different parameters. The parameters used are "*x*" which represents delay time, "*y*" represents the integrity of data maintained at the resource while executing the task, "*z*" represents the accuracy of the results, and other considerable parameters are represented by "*o*". All the parameters are quantitative parameters with values either 0 or 0.5 or 1. For the parameter "*x*", delay time gives a quantitative value 1 for a fast response, 0 for slow response and 0.5 for average response. For the parameter "*y*", if the data integrity is maintained by the resource then it is 1, if it is suspicious then 0.5, otherwise it is 0. For the parameter "*z*", if the results received are accurate, it is 1, otherwise it may be average 0.5 or not accurate 0. Therefore feedback *Fc* by client *C* is calculated as

$$Fc = x + y + z + o \quad (1)$$

Depending on the *Fc* value calculated (the maximum 4 or minimum 0), resources can be assigned into a trust category of either belief, unbelief or uncertain. The categorization decision depends on the sensitivity of the data, computation and urgency of the task to be performed on the resource. Thereby, the range for *Fc* is decided for placing a resource under a particular trust category. It is autonomous to the client and its policies. The feedback is sent to the grid resource allocation manager (GRAM).

3.2.2. Calculation of Trust

In this, the trust on a resource is calculated and kept on the recommendation table and the feedback history table.

"*Rh*" are invocative records in the recommendation table. Each grid client "*C*" holds the opinion about the trustworthiness of a specific resource "*R*" whose owner is "*Co*" which is managed by GRAM. "*Rh*" record structure is represented by the following tuple.

$$Rh = (Co, R, F, C)$$

Co is the owner of the resource, R is the resource for which trust is calculated, F is the feedback sent by the client C which used the resource recently.

The tuples or records related to *Rh* are maintained in recommendation table. In **Table 1** recommendation table, the 3 resource records and their recent feedback given by clients are shown as an example.

"*Fh*" holds the historical feedback records of the resources in the feedback table. The feedback is collected from different clients when they accessed the resources in the past. "*Fh*" record structure can be represented by the following tuple.

$$Fh = (Co, R, (C1, Qf1), (C2, Qf2), (Cn, Qfn))$$

Co is the owner of the resource R, and $(C1, Qf1), \cdots, (Cn, Qfn)$ *are the respective client's historical feedback about the resource which is given in the quantitative form. Qf* = {1, 0.5, 0} *which are the measures for be-*

Table 1. Recommendation table.

Owner (*Co*)	Resource (*R*)	Trust Degree (*F*)/ Recommendation	Updated By (C)
*C*1	*R*1	Belief	*C*2
*C*2	*R*2	Belief	*C*1
*C*3	*R*3	Unbelief	*C*1

lief, uncertain and unbelief.

The tuples or records related to *Fh* are maintained in feedback table. In **Table 2** feedback table, 3 resource records and their feedback history received from different clients are shown as an example.

The trust for a resource (*Tr*) is calculated as follows:

$$Fcs = \sum_{i=0}^{s(c)} Qfi + F \tag{2}$$

$\sum_{i=0}^{s(c)} Qfi$ *is a summation of quantitative values Qf of feedback of each client on resource taken from Fh record of a feedback table whose trust degree is being calculated i.e.* $Qf1 + Qf2 + \cdots + Qfn$.

s(*c*) *represents the number of historical clients whose feedback is maintained in the record of the feedback table Fh for the resource R. F is the recent feedback of the resource R retrieved from recommendation table.*

Fcs is a summation of feedback's of all historical clients of a resource taken from feedback table and a recent client's feedback of a resource taken from recommendation table.

Update Feedback table for the resource R with new client feedback by appending (*Cnew, Fnew*) *pair.*

$n = s(c) + 1$ *i.e. total number of historical clients* + 1 *recent client recommendation*

$$Tr = Fcs/n \tag{3}$$

n represents the total number of clients whose feedback is considered while calculating Fcs.

Whenever a Grid client wants to access a resource or inquires about the trustworthiness of the resource, the trust result for that resource *Tr* is calculated.

4. Simulation

The implementation of the proposed approach for calculating trust degree is done using Java, JDK 1.6. It consists of 3 phases.

Phase 1 is referred as Generation phase. In this grid users where each user is assigned with one gridlet/job and resources each with one machine with one processing element are generated.

Phase 2 is referred as feedback collection phase. In this a resource is assigned to a grid user for executing its job. After completing the execution using the response time a feedback is generated using Equation (1). This feedback is considered as a recommendation about the trust degree on this resource by the grid user. It is maintained in recommendation table.

Phase 3 is referred as a trust degree calculation phase. In this, the feedback table is used to find the historical feedback about a resource. This feedback table maintains the history of feedback received from various clients when the same resource is allocated to them. Using the historical record of feedback table and the current feedback of recommendation table of a resource, the trust degree is calculated using Equation (3) and Equation (4).

4.1. Illustration of Proposed Approach

The following example scenario illustrates the proposed work.

Let <owner, resource> is the pair for representing an owner of a resource in a Grid which can be shared by a user or client. Owner of a resource can become the client for the other resource owned by another owner. Let <*C*1, *R*1>, <*C*2, *R*2> and <*C*3, *R*3> are the 3 pairs of users and the resources owned by them.

Let *C*1 trusts *C*2 for using the resource *R*2 owned by *C*2 and *C*2 trusts *C*3 for using the resource *R*3 owned by *C*3. Then, according to the associative rule, *C*1 can trust indirectly *C*3 for sharing the resource *R*3 in the future. In this *C*1 should evaluate the recommendation from *C*2 about the trust on *R*3 which is owned by *C*3. *C*1 will combine its opinion with *C*2 and *C*2's cooperation about making his resource *R*2 sharable in the grid. This will

Table 2. Feedback table.

Owner (*Co*)	Resource (*R*)	History (*C*, *Qf*) pairs
*C*1	*R*1	(*C*2, 1), (*C*3, 0.5)
*C*2	*R*2	(*C*1, 0.5), (*C*3, 0)
*C*3	*R*3	(*C*2, 0.5), (*C*1, 0)

determine *C*1's opinion about the *C*2's capability as recommender.

The following steps are taken to create a trust relationship between the owner of a resource and the client/user of a resource.

1) When a client *C*1 accesses a resource *R*2 owned by *C*2 in a Grid, depending on the time taken to perform the task on the resource, delay time and accuracy of the results received, it assesses/creates a trust degree either as belief, unbelief or uncertain using the trust assessment method described above and sends it as a feedback to resource allocation manager which stores it in a recommendation table maintained by it. The feedback is calculated using Equation (1). Let the response delay is average (quantitative value is 0.5) and data integrity is maintained (quantitative value is 1) with accurate results (quantitative value is 1). The other parameters are assumed satisfactory (quantitative value is 1). Therefore, feedback of client *C*1 can be calculated as

$$Fc1 = 0.5 + 1 + 1 + 1 = 3.5$$

The client *C*1 has categorized it to "belief" by applying its policies and returns this value to allocation manager which stores it as a record <*C*2, *R*2, belief, *C*1> in recommendation table.

2) The client *C*3 can access and built its trust in the resource *R*2 by retrieving the degree of recommendation from the recommendation table and feedback table. Trust on *R*2 is calculated using the Equation (2) and Equation (3). Let the trust on *R*2 in the feedback table is present in the record with values <*C*2, *R*2, (*C*1, 0.5) (*C*3, 0)>. The information in the record says that the *R*2 resource owner is *C*2. Client *C*1 and client *C*3 have given their feedbacks about the resource in the past as uncertain (0.5) and unbelief (1) respectively. The calculation of trust degree using Equation (2) and Equation (3) is as follows.

$$Fcs = 0.5 + 0 + 1 \quad \text{(here 1 is the quantitative value of belief set by } C1\text{in step 1)}$$

$$Tr = 1.5/3 = 0.5 \qquad (\textit{i.e.}\ \text{uncertain})$$

This trust degree is used as a parameter while allocating a resource to *C*3. If no other resource is available with the trust degree of belief, then *R*2 is allocated to *C*3.

3) Depending on improved or reduced performance of resource *R*2 towards executing the task of client *C*3, the client *C*3 can increase or decrease the degree of recommendation for resource *R*2, in the recommendation table by sending its feedback to resource allocation manager by using the Equation (1). The feedback table is updated with the new client's feedback for the resource. This will give the opportunity to other clients or users in grid while taking the decision of trusting the resource *R*2.

4.2. Performance

The simulation results show that while taking the decision of appropriate resource allocation to a user, if the level of trust or trust degree is considered, then there will be an increase in the usage of the resource. It is observed that the trusted resource is frequently selected by the allocation manager while allocating the resource to the grid client. Hence, the resource can be utilized to its maximum without making it to sit idle, which satisfies the key feature of the grid environment. The following graph shows a gradual increase in the usage of the resource when its trust degree is increasing. The trust degree of a resource increases when it takes less delay time to give a response by maintaining the integrity of data and by providing accurate results.

From the results shown in the graph in Figure 2, it can be observed that there is a gradual increase in the usage of a resource when trust degree is considered while allocating the resource.

5. Conclusion

This paper gives the novel solution for maintaining a trust relationship among resource users and the resource

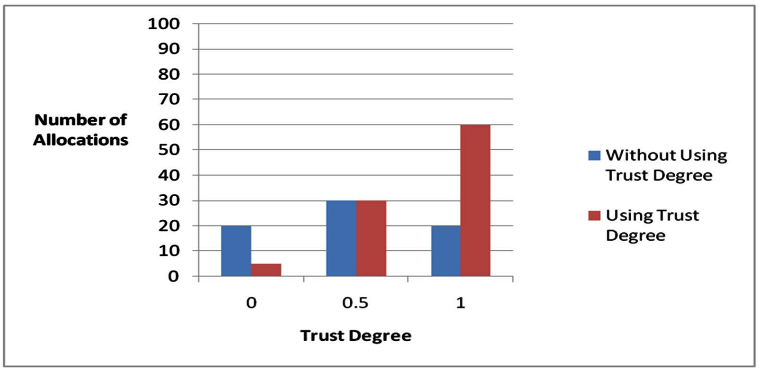

Figure 2. Comparison of allocation of a resource with and without using trust degree.

providers in the grid. The given solution increases the possibility of utilizing the trusted resource to its maximum extent. The simulation is done for assessing and recommending the trust on a resource in the form of feedback by the resource user and for calculating a trust degree of the resource depending on the feedback history of the resource. The simulation results show that there is an improvement in the selection of the appropriate resource by the allocation manager when the trust degree of a resource is considered in addition to the other parameters of a resource. Rather than depending only on digital certificates for authenticating the resources, if a trust degree of a resource is used, it results in improving the grid user satisfaction by increasing the trust on the resource on which the job is executed. In this only crisp value 0 or 0.5 or 1 are considered to the trust degree of a resource which represents unbelief, uncertain and belief. The future work can use fuzzy set theory by considering the trust degree values in the interval of real numbers [0, 1] and also approximation can be done using rough set theory.

References

[1] Foster, I., Siebenlist, F., Tereche, S. and Welch, V. (2002) Security and Certification Issues in Grid Computing. Argonne National Laboratory, University of Chicago, Chicago.

[2] Welch, V., Siebenlist, F., Foster, I., *et al.* (2006) Security for Grid Services. Argonne National Laboratory, University of Chicago, Chicago.

[3] Anitha, A. and Mujumdar, C. (2013) Hybrid Security Techniques for Grid/Cloud Computing Environment Using Identity Based Cryptography Advances. *International Journal of Computers*, *Electrical and Advanced Communications Engineering*, **2**,

[4] Foster, I., Kesselman, K. and Tzudik, G. (1998) A Security Architecture for Computational Grids. *Proceedings of the* 5*th ACM Conference on Computer and Communication Security*, San Francisco, 83-92. http://dx.doi.org/10.1145/288090.288111

[5] Grimshaw, A.S., Humphrey, A.S. and Natrajan, A. (2004) A Phylosophical and Technical Comparison of Legion and Globus. *IBM Journal of Research and Development*, **48**, 233-254. http://dx.doi.org/10.1147/rd.482.0233

[6] Cody, E., Sharman, R., Atkar, N., Rao, R.H. and Upadhyay, S. (2008) Security in Grid Computing: A Review and Synthesis. *Decision Support Systems*, **44**, 749-764. http://dx.doi.org/10.1016/j.dss.2007.09.007

[7] Kamvar, S.D., Schlosser, M.T. and Molina, H.G. (2003) The Eigen Trust Algorithm for Reputation Management in P2P Networks. *Proceedings of the* 12*th International World Wide Web Conference*, Budapest, 20-24 May 2003, 12 p.

[8] Roxburgh, A., Pawlikowski, K. and McNickle, D.C. (2004) Grid Computing: the Current State and Future Trends (in General and from the University of Canterbury’s Perspective), University of Canterbury, Christchurch.

[9] Wolski, R., Brevik, J., Krintz, C., Obertelli, G., Spring, N. and Su, A. (1999) Running Every Ware on the Computational Grid. *Proceedings of the* 1999 *ACM/IEEE conference on Supercomputing*, Article No. 6, ISBN: 1-58113-091-0. http://dx.doi.org/10.1145/331532.331538

[10] Krauter, K., Buyya, R. and Maheswaran, M. (2001) A taxonomy and Survey of Grid Resource Management Systems for Distributed Computing. *Software*: *Practical Experience*, **32**, 135-164. http://dx.doi.org/10.1002/spe.432

[11] Mattmann, C.A., Garcia, J., Krka, I., Popescu, D. and Medvidovic, N. (2015) Revisiting the Anatomy and Physiology of the Grid. *Journal of Grid Computing*, **13**, 19-34. http://dx.doi.org/10.1007/s10723-015-9324-0

[12] Srinivasan, R. and Srivaramangai, P. (2010) A Comprehensive Trust Model for Improved Reliability in Grid. *International Journal of Computer Application*, **5**, 1-4.

[13] Meng, W., Xia, H. and Song, H. (2009) A Dynamic Trust Model Based on Recommendation Credibility in Grid Domain. *International Conference CiSE*, Wuhan, 11-13 December 2009, 1-4. http://dx.doi.org/10.1109/cise.2009.5363348

[14] Farouk, A., Fouad, M.M. and Abdelhafez, A.A. (2014) Cross Domain Identity Trust Management for Grid Computing. *International Journal of Security*, *Privacy and Trust Management*, **3**, 11-21.

[15] Ashrafijoo, B., Navin, A.H., Mir Nia, M.K. and Abedini, S. (2010) Trust Management in Grid Computing Systems Based on Probability Theory. 2nd *International Conference on Educational Technology and Computer* (*ICETC*), **4**, 316-320. http://dx.doi.org/10.1109/icetc.2010.5529674

[16] Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M. and Werthimer, D. (2002) Seti@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, **45**, 56-61. http://dx.doi.org/10.1145/581571.581573

[17] Gong, L. (1997) Java Security: Present and Near Future. *IEEE Micro*, **17**, 14-19. http://dx.doi.org/10.1109/40.591650