Journal of Computer Science 7 (8): 1237-1243, 2011
ISSN 1549-3636

A Hybrid Model for Load Aware Trust Management in Grid

[1]K. Selvi and [2]R.S.D. Wahida Banu
[1]Department of Information Technology,
Paavai College of Engineering, Namakkal, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering,
Government College of Engineering, Salem, Tamil Nadu, India

Abstract: Problem statement: Trust management in a distributed dynamic environment like grid has been very vital since the allocation of appropriate resources to meet user request influences the success of the system. While considering a set of resources which are loaded invariably, balancing the load also contributes for the efficient resource utilization. **Approach:** This study proposes a trust calculation model, which considers weighted parameters like direct trust, reputation trust, load average information and network efficiency. Network efficiency also varies with the granularity of jobs, hence the experiment were conducted with an inclusion of this criteria. **Results and Conclusion:** The proposed Hybrid Model for Load aware Trust management system (HMLT) allocates the job based on the trust values. The resources with higher trust value get more jobs. As a result the performance of the proposed system found to be faster and has a better utilization of grid resources when studied in a grid environment.

Key words: Trust management, network efficiency, load index, grid environment, separate services, grid resources, trust calculation, behavioral trust

INTRODUCTION

Grid is a distributed computing technology that supports aggregation of distributed computational resources that span beyond organizational boundaries. The coordinated use of resources meets the requirements of advanced science and engineering. Grid can be distinguished from conventional distributed computing by its focus on large scale resource sharing, high performance and solving compute/data intensive applications. Grid supports researchers and scientists from diverse organizations to share information, instruments, data, compute and storage resources dynamically in a flexible and secure manner(Foster *et al*., 2001) thereby forming a 'Virtual Organization' (VO) to solve challenging applications. The resources in grid are dynamic and are organized as a number of virtual organizations. The real complexity in scheduling a user job lies in identifying the suitable resource with the expected capability of the user job execution need. The process of match making in the domain of VOs becomes much critical due to the dynamic availability of resources in a grid. This led to the introduction of the familiar methods of adding the notion of trust with grid resources. Trust in the context of grid is classified broadly into security trust and behavioral trust; the former refers to the means of available protections for securing resources and the latter is concerned with the expected performance of the resource. Trust in traditional terms cannot be measured with any other domain where it is applied. In grid and web, the trust models work with trust calculation and trust is equated to an integer. In the proposed model, the behavioral trust of grid resources was considered. Behavioral trust is usually calculated by combining direct trust which is evaluated with the direct contacts of resources with the reputation trust that are received from other well known contacts' recommendation. Further, the trust calculations are normally made taking into consideration the decay of the calculated trust over a time since the environment being dynamic and the trust value changing often.

Trust calculation techniques developed so far have not taken into account the load balancing, job size and network efficiency which are the significant parameters that influence performance. Hence, the proposed model discusses a novel trust calculation paradigm in grid considering parameters such as job granularity, network efficiency and past behavior of resources. These parameters are measured through separate services, which together can be exposed as a single web service interface. The use of trust in the proposed model distributes the load based on the trust values. The resources which have higher trust value get more

Corresponding Author: K. Selvi, Department of IT, Paavai College of Engineering, Pachal, Namakkal, Tamil Nadu, India
Tel: +91 99427 33014

number of jobs. Hence the job completion will be approximate to the user specified time. The trust cannot be calculated exactly since it is a belief; thereby this model approximates the trust calculation into a mathematical equation by taking the above mentioned parameters.

Related work: Trust is defined as the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subject to the entity's behavior and applies only within a specific context at a given time (Ma *et al*., 2006; Azzedin *et al*., 2006). The firm belief value will be in the range of maximum and minimum trust values. The reputation of an entity is an expectation of its behavior based on other entities' observation or information about the entity's past behavior within a specific context at a given time. The related works on trust metric calculation are discussed in detail to capture the state of art in trust and reputation management.

Vivekananth (2010) the behavior based trust model for resource selection was discussed. The trust calculation depends on penalty, feedback from past experience, context and time. The value of the penalty may vary from 0 to 1, based on the harm created by the misbehavior resource. Chen *et al*. (2009) proposed an approach for resource allocation and selection. In their work they considered trust values along with the local strategies of the resources.

The (Liu *et al*., 2008) subjective logic is a trivalent one, an opinion can have 3 degrees of values: belief (b), disbelief (d) and uncertainty (u), with $b + d + u = 1$, $\{b, d, u\} \in [0, 1]$. The main contribution of their work is a clear representation of the logic of each node in the network and also the trust degree threshold function has been designed for dependent tasks.

Vijayakumar and Banu (2008) have proposed a method for resource selection in grid environment using trust and reputation. The trust value of each entity is calculated based on self-protection capability weightage and reputation. Multi Criteria Resource Selection (MCRS) algorithm for resource selection which considers processing time, workload and bandwidth was discussed by (Malarvizhi Nandagopal and Rhymend Uthariaraj, 2011). According to (Punam Bedi and Hema Banati, 2006) different quantitative measures of user trust on a website are discussed.

The above said models have not incorporated load balance and network efficiency in the grid. So in this work these two parameters are included to optimize the trust calculation and to increase performance by giving more number of jobs to the resource which has the highest trust value.

Types of trust: Trust can be defined in many ways. In this work it is defined as reasonable expectation (confidence) of the trustor that the trustee will behave in a way beneficial to the trustor.

Identity trust is the ability of a party to determine, with some level of certainty that an electronic credential representing an entity - whether a human or a machine, with which it interacts to effect a transaction, can be trusted to actually belong to the entity.

The behavioral trust deals with a wider notion of an entity's 'trustworthiness'. In this work the behavior trust is calculated from the weighted combination of three parameters such as:

- Individual Dimension (approximation to direct trust), which is the direct trust obtained by previous experience with another agent
- Social Dimension (approximation to behavior trust), which refers to the trust of an agent in relation with a group and the ontological dimension which reflects the subjective particularities of an individual
- Client Efficiency which can be used for dynamic assessment of network efficiency and workload

Motivation and contribution: The reason for the motivation of the proposed model is the fact that previous models have not utilized the concept of dynamicity completely and have used two dimensions for trust calculation and scheduling. In contrast, this work proposed utilize fully, the concept of dynamicity and combine the trust calculation and scheduling into one dimension which makes this model suitable for grid environment. The developed model considers the present load as well as network efficiency of each grid node before assigning a job to that node. HLMT is hierarchical, extensible and seamlessly pluggable with meta-schedulers.

Trust architecture: The proposed hybrid trust management architecture takes into account different domains of Virtual Organizations (VOs). Each domain maintains a separate individual dimension trust Table consisting of trust values for every other domain within a specific context. The context considered are storage, processing capability and request forwarding. Figure 1 depicts the layout of the proposed hybrid trust management system.

The internal architecture of the trust parameter assessors is shown in Fig. 2. The trust value is calculated from the various modules present in this architecture which are listed below:

- Individual dimension assessor

- Social dimension assessor
- Network efficiency assessor
- Load index assessor

Individual Dimension assessor module calculates the individual dimension or direct trust from the past experiences stored in the Individual Dimension trust Table (IDT), by decaying the value in IDT based on time difference of the updated time and current time. Social dimension assessor calculates the average of reputation values from other entities. The network efficiency of the domain is the measure of data transfer speed to and from the domain and the confederation based on the relative speeds of all the other domains in the grid. Load index is the measure of current workload of the resources under consideration. It is calculated as time taken to execute a proportionate part of the whole job on the resource under consideration and thereby estimate the current load of the resource for full load of job.

Implementation and working of TMS: When the job from domain S_m arrives, the meta-scheduler calls load aware Trust Management System (TMS) which in turn calculates the trust values for every resource (site) present in the various other grid VO domains (S_1,S_2..S_n). After calculating the values of assessors present in the architecture the TMS updates these values in the IDT of S_m. Then the meta-scheduler assigns the job to resources with most trust value at specific context.

Trust calculation: The various calculations that are evaluated by the constituent modules presented in the architecture are explained subsequently. Since the value of trust cannot be calculated exactly, this model takes into account some of the parameters to determine the approximate trust value. The trust values are constricted to be in the range from 0 to 100. Let S_m and S_n denote two domains of entities. The trust relationship based on a specific context c at a given time θ between the two domains is expressed in the equation as the weighted combination of ID, SD and CE:

$$T(S_m, S_n, \theta, c, g) = \alpha \times ID(S_m, S_n, \theta, c) + \beta \times SD(S_m, S_n, \theta, c) + 0.5 \times CE(S_n, \theta, g)$$

Where:

ID = Individual dimension of trust.
SD = Social dimension of trust.
CE = Client efficiency.
g = Granularity of job.
α, β = Weights given to trust factors ID and SD respectively:

And $\alpha + \beta = 0.5$

Individual dimension calculation: Trust value in Individual Dimension Trust Table (IDT) decays with time due to dynamicity. Hence in this account, the individual dimension is taken to be a product of value in IDT and decay function (Azzedin *et al*., 2006):

$$ID(S_m, S_n, \theta, c) = IDT(S_m, S_n, \theta, c) \times \lambda(\theta - \theta_{mn,} c)$$

Where:

$IDT(S_m, S_n, \theta, c)$= Trust value for a specific context c and domain S_n in IDT maintained by S_m
$\lambda(\theta - \theta_{mn}, c)$ = Decay function

Significance of decay function: As any other relationship, trust decays with time. For instance, if S_m has not interacted with S_n for a longer duration, then the current trust T between them is likely to be weaker unless they have interacted recently. Hence, the trust model introduced here employs a decay function to reflect this drop when modeling trust between domains. The time difference that resulted from the last transaction between S_m and S_n and the current time are taken to compute the decay function $\lambda(\theta - \theta_{mn}, c)$. Each domain might have different decay function and might be looking at other factors that accelerate or decelerate the trust decay.

Social dimension calculation: Social dimension is calculated as the average of reputation value obtained from other entities (Azzedin and Maheswaran 2006). The mathematical equation for reputation trust is given as:

$$SD(S_m S_{n,} \theta, c) = \frac{\sum_{k=1}^{n} IDT(S_k, S_n, c) \times RF(S_m, S_k, c) \times \lambda(\theta - \theta_{kn}, c)}{\sum_{k=1}^{n} S_k}$$

Where:

$IDT(S_k, S_n, c)$ = Trust value for a specific context c and domain S_n in IDT maintained by S_k
$RF(S_m, S_k, c)$ = Recommender factor of S_k maintained by S_m
$\lambda(\theta - \theta_{kn}, c)$ = Decay function

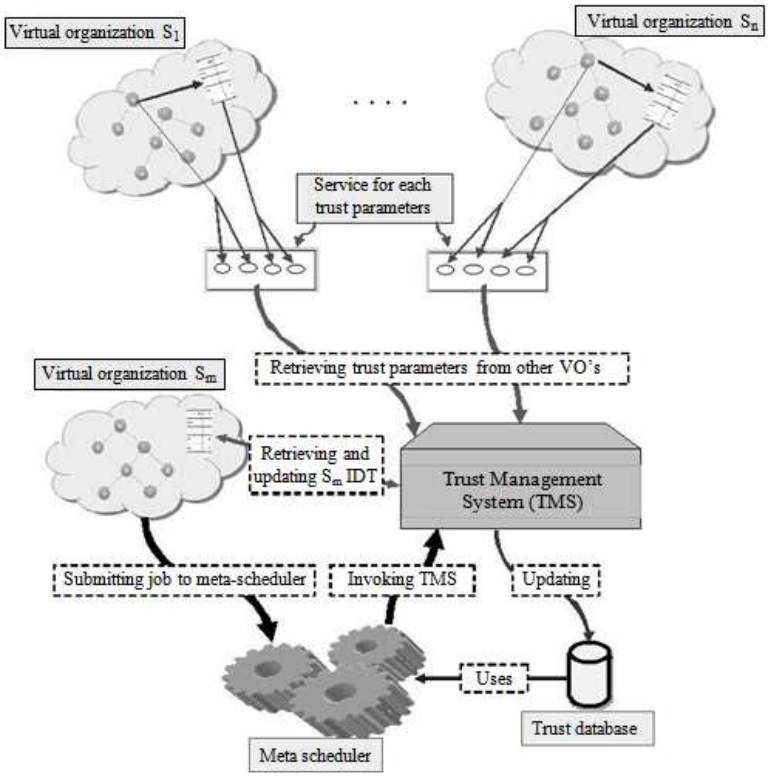

Fig. 1: Proposed hybrid trust management architecture

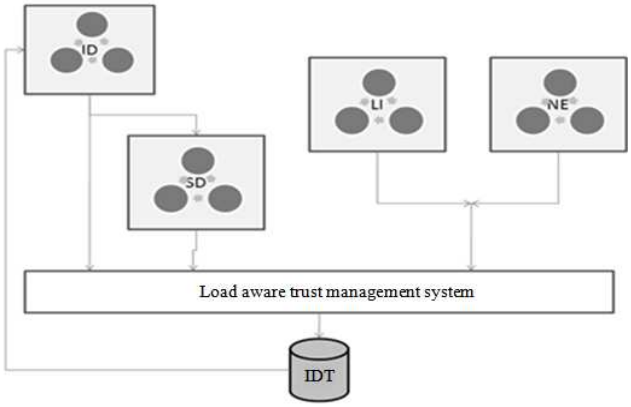

Fig. 2: Internal architecture of the trust parameter assessors

Table 1: Individual Dimension trust Table maintained by S_n

	Sites					
Context	-------S_1--------				--------S_m------	
	T	R	⋯	⋯	T	R
c_1	$T_{1,1}$	$R_{1,1}$	⋯	⋯	$T_{m,1}$	$R_{m,1}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
c_i	$T_{1,i}$	$R_{1,i}$	⋯	⋯	$T_{m,i}$	$R_{m,i}$

Recommender trust factor: Reputation is based primarily on what other domains say about a particular domain. The recommender trust factor RF is introduced in the trust model to prevent cheating via collisions among a group of domains. Also, RF is a value between 0 and 1 and will have a higher value if S_m and S_k are unknown or have no prior relationship among each other and a lower value if S_m and S_k are allies or business partners.

Individual dimension trust table: Each domain maintains an Individual Dimension trust Table (IDT) whose structure is illustrated in the Table 1. From this Table it is seen that for a specific context c, every site S_n maintains the trust and recommender factor value for every other site present in the domain represented in the columns $S_{1,}$ $S_{2,..}$ S_m.

To update the IDT, the following equation is used:

$$IDT(S_m,S_n,c) = \tanh ((T (S_m,S_n,c)+ \Delta) \times IDT (S_m,S_n,c))$$

Where:

$T (S_m,S_n, c)$ = The trust value for context c resulted from the direct trust relationship between S_m and S_n at time θ.

$IDT(S_m,S_n, c)$ = The trust level in the IDT for context c resulted from the last direct transaction between S_m and S_n.

Δ = A value between 0 and 1. If $\Delta > 0.5$, higher preference is given to T resulting from the current direct trust relationship between the two domains

The required trust value is defined as a value between 0 and 100, such that:

if $T(S_m,S_n, \theta,c,g) >= RTv$, the interaction is trusted and the request is granted

if $T(S_m,S_n, \theta,c,g) < RTv$, the interaction is not trusted

Client efficiency calculation: The value for client efficiency is calculated from the following equation:

$$CE(S_n, \theta,g) = g \times LI (S_n, \theta) + (1-g) \times NE (S_n, \theta)$$

Where:

$LI (S_n, \theta)$ = load index of S_n at θ

$NE(S_n, \theta)$ = network efficiency of S_n at θ

Load index calculation: The load index (Ahmed *et al*., 2008; Nandagopal and Uthariaraj, 2011) depends on the job complexity. So complexity of the job and its computation is required before the load index calculation. This is calculated using the following algorithm:

- Take the size of the whole data set provided by the user as n. Take a small part (1-10) percentage of the whole data set as a. Calculate time to execute 'a' part of the whole job on local resource and estimate time to execute 100 % of the job(T_a)
- Obtain time constraint (Total time in which the job must be completed) T_n from the user
- Calculate the load index of the individual processing element using T_a and T_n

After calculating the complexity, the load index (LI) is calculated which is the measure of workload of the processor. It is in the scale of 0-100. If the processor is busy with many jobs the load index will be low. It will be inversely proportional to the processor load. The LI is calculated using the formula:

$$LI = \frac{T_n \times LI_{max}}{T_a}$$

The ideal domain, which has the LI as 100, is the domain that completes the job in the user specified time. The other domains are rated with this domain as the reference.

Network efficiency calculation: The network efficiency of the domain is the measure of its data transfer speed and the confederation based on the relative speeds of all the other domains in the grid (Nandagopal and Uthariaraj, 2011). It is in the scale of 0 to 100. The domain in the grid with the longest transfer latency is given a network efficiency of 0 and a local domain is given a network efficiency of 100. All other domains are given intermediated values related to these domains. Thus, the network efficiency is inversely proportional to the latency time. The network efficiency is calculated using the following algorithm which is specific for the proposed model:

- Domain D_i broadcast an 'Enquiry' message to all domains and note down the time of broadcast of each message
- When a domain receives an 'Enquiry' message, it calculates the current processor load on the domain and sends it back as a 'Status' message back to domain $D_{i.}$ Based on the message timestamps the domain D_i calculates the round trip latency RTL. The network efficiency (NE) is calculated as follows:

$$NE = 1 - \frac{Lat_i}{Lat_{max}}$$

Where:

Lat_i = Round trip latency of the domain

Lat_{max} = Maximum of the all domains in the grid

MATERIAL AND METHODS

The proposed model uses web services model for the trust metric calculation and the load distribution method while distributing the workload among the grid nodes. The application of key strength determination employs the brute force method of key combination.

RESULTS AND DISCUSSION

To evaluate the performance of the implemented HMLT, a key strength determining application is tested with 12 domains of VOs. This experiment is performed in a grid environment created depicting the VOs. The TMS and meta-scheduler are also modeled in this environment. The brute force attack of various possibilities (key lengths-3, 4….n) have been given to the resources. The number of combination given to the resources depends on the trust value. Hence, the highest trust value of the resources will get more jobs than others. The time taken to break the password when the application was executed in standalone systems, traditional grid system and the implemented grid with HLMT are presented in Table 2.

As the Job granularity increases with network efficiency as constant, more importance is given to load index which is shown in Fig. 3 i.e., when executing a job of large granularity, minimal resources with high trust values and which are lightly loaded are selected, on the other hand when the granularity is small the job is evidently split and executed simultaneously over a set of resources with moderate load and trust values. The plot of network efficiency against client efficiency for various job granularities is shown in Fig. 4. When scheduling coarse grained job, client efficiency (network efficiency) is negligible, alternately when dealing with a fine grained job, client efficiency is taken to be significantly with a higher weight.

The direct trust influences the ultimate final trust value calculated i.e., when direct trust value increases the trust calculated also yields to be higher. The variation of alpha values with direct trust and trust is shown in Fig. 5.

The plot of various trust components is shown in Fig. 6. From the plot it is clearly inferred that the efficiency of the system is high when considering trust with load and network efficiency i.e., the job completes without fail and in less time when the proposed trust model is considered.

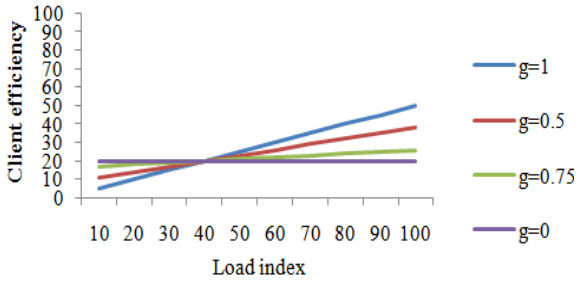

Fig. 3: Relation between load index, client efficiency and job granularity

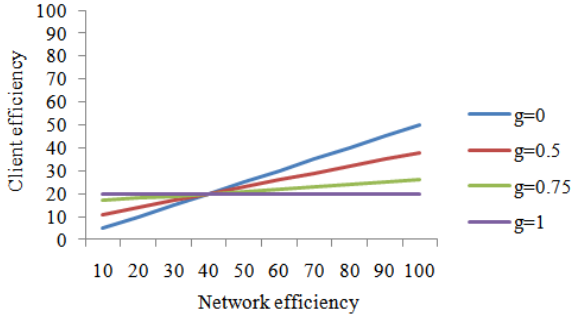

Fig. 4: Relation between network efficiency, client efficiency and job granularity

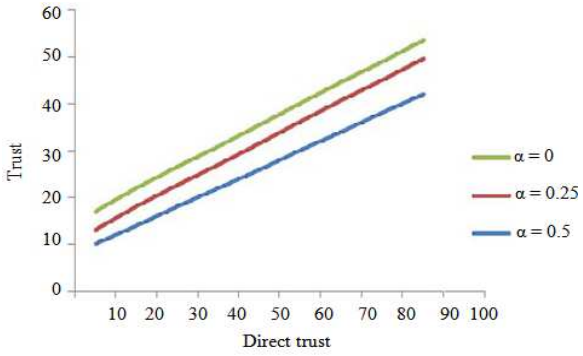

Fig. 5: Relation between trust and direct trust for various alpha values

Table 2: Time taken to break the password

		Time required		
Password length	No of possibilities	Stand alone system (ms)	Grid system (ms)	Proposed grid system with HMLT (ms)
3	456533	62	70	68
4	35153041	78	79	76
5	2706784157	86	84	80
7	1.6×10^{13}	112	101	92
10	7.3×10^{18}	186	162	149
15	2.0×10^{28}	277	249	210

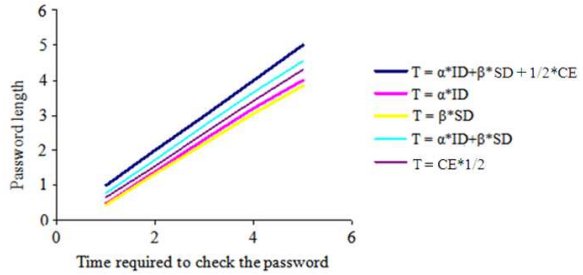

Fig. 6: Comparison of various trust components

CONCLUSION

When the concept of trust being incorporated in grid, the performance of system increases. The usage of trust in grid makes the system more reliable where the job is executed within user constrained time. There results increase of grid usage with trust system rather than grid without trust. Also, in the proposed model various trust modules are being exposed as separate services; the flexibility in using the grid increases. The inclusion of load awareness further improves the effective utilization of resources in the grid.

REFERENCES

Ahmed, B.S., K. Samsudin and A.R. Ramli, 2008. Architectural review of load balancing single system image. J. Comput. Sci., 4: 752-761. DOI: 10.3844/jcssp.2008.752.761

Azzedin, F., M. Maheswaran and A. Mitra, 2006. Trust brokering and its use for resource matchmaking in public-resource grid. J. Grid Comput., 4: 247-263. DOI: 10.1007/s10723-006-9041-9

Bedi, P. and H. Banati, 2006. Assessing user trust to improve web usability. J. Comput. Sci., 2: 283-287. DOI: 10.1.1.165.7336

Chen, C., G. Li-ze, N. Xin-xin and Y. Yi-xian, 2009. An Approach for Resource Selection and Allocation in Grid Based on Trust Management System. Proceedings of the 1st International Conference on Future Information Networks, Oct 14-17, IEEE Xplore, Beijing, Chaina, pp: 232-236. DOI: 10.1109/ICFIN.2009.5339585

Foster, I., C. Kesselman and S. Tuecke, 2001. The anatomy of the grid: Enabling scalable virtual organizations. Int. J. High Perform. Comput. Appli., 15: 200-222. DOI: 10.1177/10943420010150030222

Liu, Q., Y. Liao, B. Tang and L. Yu, 2008. A trust model based on subjective logic for multi-domains in grids. Proceedings of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Dec. 19-20, Wuhan, 2, pp: 882-886. DOI: 10.1109/PACIIA.2008.152

Ma, B., J. Sun and C. Yu, 2006. Reputation-based trust model in grid security system. J. Commun. Comput., 3: 41-46. DOI: 10.1.1.132.8438

Nandagopal, M. and R. Uthariaraj, 2011. Performance analysis of resource selection algorithms in grid computing environment. J. Comput. Sci., 7: 493-498. DOI: 10.3844/jcssp.2011.493.498

Vijayakumar, V. and R.S.D. Wahidhabanu, 2008. Trust and reputation aware security for resource selection in grid computing. Proceedings of the International Conference on Security Technology. Dec. 13-15, IEEE Xplore, Hainan Island, China, pp: 121-124. DOI: 10.1109/SecTech.2008.46

Vivekananth, P., 2010. A behaviour based trust model for grid security. Int. J. Comput. Appli., 5: 1-3. DOI: 10.5120/922-1300