International Journal of Soft Computing 2 (4): 494-501, 2007

Fuzzy Based Resource Management in Broker Architecture Using Trust and Credentials

P. Varalakshmi, S. Thamarai Selvi, P. Kanchana, J. Ramesh and V. Pavithra
Department of Information Technology, MIT, Anna University, Chennai 600044, India

Abstract: Multi-broker grid environment gives very high importance to the concept of trust. This architecture consists of entities (Consumers and Service Providers (SP)) spread across the brokers with each SP connected (logically) to more than one broker and the consumer is free to contact a broker of its choice, when it requires a service. In this study, we propose to introduce a trust management architecture that not only supports a choice of SP based on reputation (trust-index) but also on credentials (policy).Thus trustworthy SP is assigned to a consumer that not only satisfies the consumer's requirements but also its policy constraints. Thereby the consumer is assigned with a trustworthy SP and the transaction is free from runtime failure as the policies have been matched. The concept of registration of an SP to a broker has been dealt with. The concept of de-registration of an SP, has been introduced in two scenarios, one is an SP deregisters by choice and the other is the broker forcing deregistration of an SP. Publishing of brokers has been dealt with based on SOA. The SP's trust-index is updated based on both the consumer's feedback and the broker's feedback. SP's trust-index is updated periodically instead of after every transaction. The time for periodic updation of the trust-index of SP is set proportional to SP's current trust-index. This helps to reduce the brokers' workload and the number of messages exchanged The consumer's trust-index is updated based on feedback from the SP. The model proposed allows the consumers to assign priorities to parameters while requesting for a service. Fuzzy logic has been incorporated to calculate the trust-index of SP in our model. This ensures the use of fuzzy inferences in the system which can handle any imprecise linguistic terms, in this case trust-index calculation effectively. This model shows a marked improvement in cost-loss reduction to the consumers and reduction in the number of messages exchanged in the environment.

Key words: Policy-based, trust-index, presence-index, broker, SOA

INTRODUCTION

Grid computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed domains. This environment has two kinds of entities. One is the consumer that requests for services and the other is the SP that provides the service. In this scenario, for an SP to allow its services to be used and a consumer to accept the service, a honorable relationship based on faith has to be established. This can be done based on two aspects.

One aspect is the credentials (Varalakshmi *et al*., 2007). Credential is a subject relating to third party accreditations that an entity has acquired. This is a direct way where by a consumer concludes on the worthiness of an SP purely based on his credentials (policy-based system). The weight given to each policy depends on the third party that has provided the accreditations.

The other aspect considered is the trust-index (Azzedin and Maheswaram, 2002, 2004; Varalakshmi *et al*., 2006). Which incorporates honesty, truthfulness and competence of the entity. This trust-index factor is established for an SP based on past transactions. Any consumer who has a transaction with an SP provides a feedback about the SP's service. These values are aggregated over a period of time for an SP. A consumer first verifies the past behavior of an SP based on these feedbacks, before using its service.

Both these aspects help encouraging belief between two entities that are strangers and dissuade participation by those who are dishonest.

Fuzzy logic is employed where there is need to mimic the human way of reasoning. The rule structure of a fuzzy inference system makes it easy to incorporate human expertise about the application directly into the modeling process. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy

Corresponding Author: P. Varalakshmi, Department of Information Technology, MIT, Anna University, Chennai 600044, India

logic. Since trust, satisfaction are not absolute values but continuous terms, they go well with uncertainties. Thus the fuzzy system is incorporated in the trust-index computation of SPs in our model.

A domain with single broker is proposed in Varalakshmi *et al.* (2006). This results in a single-point failure. The model proposed in Varalakshmi *et al.* (2006) assigns the initial trust-index value of SP randomly. In our suggested model, single point failure is avoided by expanding the architecture to a multi-broker environment; also we evaluate the initial trust-index of SP based on the presence-index. The reputation-based system has been proposed (Azzedin and Maheswaram, 2002, 2004; Varalakshmi *et al.*, 2006) and the credential (or policy) based system has been suggested in Del *et al.* (2005) and Ionut *et al.* (2005). In our model we integrate both these factors as in Piero *et al.* (2005) to ensure trust-worthy entities are participating in a runtime failure free transaction. The SP's trust-index updation after each transaction and false feedbacks detection were proposed in Azzedin and Maheswarah (2004), Varalakshmi *et al.* (2006). In our model the broker monitors and verifies the trust-index at specific instances of time to reduce message overhead in the environment. In the model recommended in Azzedin and Maheswaram (2002, 2004), Varalakshmi *et al.*(2006) the entities associate themselves with any broker randomly. The main drawback of this system is that the trust-worthiness of the entities can only be predicted after association. In our model, the trust-worthiness of an entity is verified before the broker allows any association with that entity. The architecture in Varalakshmi *et al.* (2007) proposes the concept of broker-forwarding request whereby once the broker obtains a request from a consumer, it not only processes the request itself but also forwards the request to all other brokers it is associated with. This increases the overhead at the broker site. Our model follows the SOA whereby the broker on obtaining a request from the consumer simply processes it and provides a reply. If the broker is unable to find an SP, the consumer is free to send the same request to another broker from the list of brokers available in a common site. This continues till it obtains an SP. A fuzzy-based trust-index model that considers satisfaction-index based on consumer feedback has been proposed in Samia *et al.* (2005), Shanshan *et al.* (2005), Erika and Rene (2007). However, our model considers the effect of satisfaction values (feedback) provided by both the consumer and the broker.

PROPOSED ARCHITECTURE

In multi-broker per domain environment the entities are spread across the brokers as shown in Fig. 1.

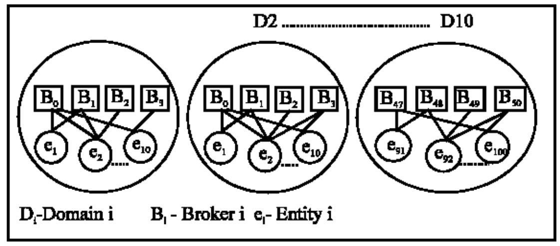

Fig. 1: Broker architecture

The brokers maintain information about the SP connected (logically) to them. The advantage of this multi-broker architecture over single broker architecture (Azzedin and Maheswaran, 2002; Varalakshmi *et al.*, 2004) is avoiding single point failure. This architecture also proposes for one SP to be attached to at least two brokers in the same domain, this provides redundancy of the details of the SP's. This improves the reliability of the system as even if one broker becomes unavailable, the SP can still service requests from the other broker.

This architecture is implemented with an unequal, non-uniform distribution of SP's. The SP's are free to attach themselves to any broker present in their domain. Once the SP associates itself with a broker the database at the broker's site has to be updated with the details of the SP. The SP's details such as SP's ID, type of services provided, criticality factor of the service, cost per hour for using the service and the trust-index are stored at the associated broker's site. The SP also provides the details of Infrastructure (I-index), Affiliation (A-index) and Policy (P-index) during the time of publishing and registering the services at the broker-site. I-index is based on the bandwidth, the system-capability, failover process (is provided or not), business continuity and disaster recovery (BC/DR is available or not) and adherence to infrastructure standards (yes or no). A-index includes the information security practices certified by international standards authorities such as BITS, ISO 270001, etc. P-index includes the presence of and adherence to the policies on warranty, payment and delivery. These details are consolidated as the presence-index of the SP. The trust index value is taken to be a normalized value between 0 and 1 as stated in (Varalakshmi *et al.*, 2007).

Publishing of brokers: SOA is defined as an architecture that is made up of services that are discrete and independent and they self advertise themselves.

In our model, whenever a broker has to join a grid network, it publishes itself in the domain where it is present currently to make sure that the other entities to know of its existence, as in SOA. These brokers are included in the domain at the time of creation of brokers through publishing.

Initially, the trust-index of a broker is assigned by a standard forum such as the GGF. This trust-index along with broker id is published at a common site for the benefit of the consumer community (who wish to approach brokers for membership and services) and the SPs (who wish to approach the brokers for membership). The broker's trust-index is updated after every registration or deregistration of an SP. The trust-index is computed as the average of the trust-indices of the SPs that belong to the broker.

Registration of SP to brokers: The SPs are free to join any of the brokers present in the same domain. There is no condition or rules the SP's have, to select a broker. This selection of the broker is not forced by any external factors but on the basis of the internal factors like what the SP thinks of the broker (i.e.,) the broker's reputation. When the SP selects a particular broker, it sends a request to that broker along with some of its details. The following steps are carried out once the broker receives a request:

- The trust-index for entities is calculated based on the presence index of SP.
- The brokers shall offer membership for an SP if and only if the trust-index of that SP is greater than or equal to its own trust-index.Otherwise, membership is disallowed, irrespective of any business profit. This is done in order to maintain its trust-worthiness in the industry, since the trustworthiness of a broker is a direct function of the trustworthiness of all the SPs associated with the broker.
- Since the parameters of presence-index are authenticated by third party, the trust-index calculation can be further fine tuned.
- If the SP is allowed to be associated the trust-index of broker is recalculated.

Deregistration of SP by choice: Once the SP registers itself with a broker, it services requests, only of those consumers who approach the broker to which the SP is associated. The broker as explained previously only chooses the best SP for the consumer, hence there can be a scenario where one SP remains idle for a long time since the trust or policy or requirement constraints of the consumers approaching the broker did not match the SP or even if there is a match it is not ranked as the best SP and hence not chosen. These SP are then free to disassociate themselves from this broker and once again approach another broker for registration as explained.

To deregister the SP will send a request to the broker for deregistration. The broker then looks at the history of transactions and based on the last time the SP had a transaction and the current trust-index of an SP it either allows or disallows the SP to deregister, since the lower the trust-index the longer an SP has to remain idle.

Broker forcing deregistration of SP: When an SP continuously behaves viciously in one way or the other, it obtains a continuous set of low feedbacks from the consumers and hence its trust-index value decreases. Since the broker's trust-index is a direct function of the trust-indices of SPs associated with it, the net effect of an SP behaving viciously results in the broker's trust-index value becoming lower. Hence, if an SP misbehaves for certain amounts of time, the broker forces the SP to deregister i.e., disassociate itself from the broker. But the SP may approach another broker for registration as explained.

WORKFLOW OF ARCHITECTURE

Consumer request: When the consumer sends in a request to the broker it specifies the following parameters the consumerid, required trust-index and type of service required; budget allotted for the service, deadline and its policy requirements. Apart from this the consumer also assigns a priority to the Qos parameters like cost and deadline. So when the broker is selecting an SP for the consumer he gives preference to those SPs that satisfy the higher priority parameters of the consumer. Thus the choice of SP for that consumer is further fine-tuned in accordance to the consumer.

Calculation of satisfaction indices: When a consumer requires a specific service, it sends its request to any broker.

If the consumer is interacting with this broker for the first time, then the consumer is assigned with a median trust-index value (0.5 in this case is one way of taking the median value) and then examines its table to find out the best SP available and notifies to both the entities. The SP then supplies the service to the consumer and that transaction takes place. The transaction details are updated at the broker's site. If the broker in unable to find an SP for the request, the consumer is free to forward the same request to another broker. At the end of the transaction the consumer provides a feedback in the form of a satisfaction-index (between 0 and 1). In our architecture, the broker also provides a feedback in form of a satisfaction-index (between 0 and 1) (Varalakshmi *et al*., 2007) since the broker continuously monitors the execution of job at the associated SP's site.

The SP also provides a feedback in the form of a satisfaction-index for the consumer and the trust-index of the consumer is updated at broker's site.

Hence, if the consumer interacts with the broker again the trust-index value of the consumer updated by the broker at the end of the previous transaction is taken.

The feedbacks to SP provided in the form of a satisfaction-index (between 0 and 1) are computed as follows.

The consumer's satisfaction-index S (U_c,SP), is the weighted sum of the percentage of completion of assigned work (S1), whether the job was completed within the budget (S2), the level of commitment of the SP to minimize the schedule-variance (S3) and the level of intrusion if any, that can be detected by audit data at the consumer's site (S4) and $\tilde{a}_j$ is the weight associated with the parameters. The satisfaction-index of the consumer is given as:

$$S_i(U_c,SP) = \sum_{j=1}^{j=4}(\gamma_j * S_j)$$

Where S_i (U_c, SP) is the satisfaction-index provided by the consumer (U_c) to the SP for 'i' th transaction, U_c is the ID of consumer, Υ_j is the weight associated with the satisfaction-index parameters S_j. The broker's satisfactions-index S_i (B_{id}, SP) is computed using the following equation as:

$$S_i(B_{id},SP) = \sum_{j=1}^{j=5}(\gamma_j * S_j)$$

Where B_{id} is the ID of broker, S1-S4 are similar to the consumer's satisfaction-index parameters, S5, the presence-index, is an average of the values of I-index, A-index and P-index.

The broker then consolidates both these feedbacks as a single satisfaction-index,(S_i (U_c, B_{id}, SP) taking the average of the two values.

The SP's satisfaction-index S (SP,U_c), is the weighted sum of the percentage of utilization of assigned resource (S1), whether payment commitment was honored (S2) and Υ_j is the weight assigned to the parameters. The satisfaction-index of the SP for a consumer is given as

$$S_i(SP,U_c) = \sum_{j=1}^{j=2}(\gamma_j * S_j)$$

The updated Trust-Index value of consumer ($(TI(U_c)_{new})$) is given by following equation:

$$(TI(U_c))_{new} = \frac{(TI(U_c))_{old} + S_1(SP,U_c)}{2}$$

Where $(TI\ (U_c))_{old}$ is the last updated trust value

Integrating credential and reputation: As stated, there are 2 ways to check the genuineness of a SP, based on trust-index or by verifying the credential. In our model we propose to combine these two factors.

The consumer requests will include certain policy requirements that the SP should posses. The broker processing the consumer request tries to find an SP that not only matches the trust constraints but policy requirements. Thus the runtime failure due to policy mismatch is avoided.

Considering the SP's point of view, when they associate themselves with a broker they specify a set of policy requirements that the consumer should posses. The broker allots the SP only to those consumers who satisfy both the trust-index and policy requirements. The type of policies considered here are the third party accreditations, presence of digital certificate, some public key cryptosystem presence etc.

If (consumer's trust-index constraint matches SP's trust-index constraint).

If (consumer's policy-constraint matches SP's policy-constraint)

Broker allots the SP to the consumer

End

End

Periodic updation of SP's trust-index: The broker monitors the trustworthiness of the SP by periodical checking (not after each transactions) to maintain integrity of trust-index of the SP with the lesser broker's workload. During this periodic checking only, the trust-index of SP is recalculated and updated. The time for monitoring is set based on the current trust-index of the SP. The time for monitoring is set proportional to the trust-index of SP. One mechanism to fix the time for monitoring could be direct proportion say if trust-index is 0.7 the trust-index is updated every 70 transactions. So, in general, the lower the trust-index the broker updates the trust-index of the SP less frequently and vice versa. In the model, the broker's workload also gets reduced as there is no updation done after very transaction. The SP trust-index recalculation and updation is as follows. A timestamp is associated with every transaction, using which the broker separates the past transactions from the more recent ones. The older transactions are given a lesser weight (α) and the newer ones with a higher weight (β). The criticality factor CR_i (between 0 and 1), the value determines the nature of service (i.e.,) real-time and non real- time is taken. The normalized cost factor C_{ni} is also considered while computing the trust-index of SP. Thus the Trust-Index of SP, TI(SP) is given by the following equation as

$$TI(SP) = \begin{bmatrix} \alpha * \dfrac{\sum_{i=1}^{t_1} (S_i(U_c, B_{id}, SP) * C_{ni} * CR_i)}{t_1} \\ + \beta * \dfrac{\sum_{i=1}^{t_2} (S_i(U_c, B_{id}, SP) * C_{ni} * CR_i)}{t_2} \end{bmatrix}$$

Where Normalized cost is calculated from the actual cost (Ci) of transaction as:

$$C_{ni} = (C_i - 1) / C_i$$

FUZZY LOGIC

As mentioned, fuzzy logic is deployed to handle imprecise linguistic terms. In our model, we are making use of the Mamdani's fuzzy inference method. Mamdani-type inference uses an aggregation operator to combine consequents of each rule and a defuzzification method to defuzzify the resulting fuzzy set to obtain the output of the system. The main advantage of using this engine is its widespread use and it is well suited to human input.

A Membership Function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1.Gaussian function (gaussmf) is the function chosen for representing the membership, as both broker's and consumer's satisfaction-index is represented as a probability distribution. Gaussian function is most suited for probability and statistical distributions.

These input variables (broker's and consumer's satisfaction-index (feedback) and normalized cost) are fuzzified using membership functions and expressed as linguistic variables (Table 1). The rules from the rule base are triggered depending up on the current inputs. The final step is the defuzzification process where a crisp output value (trust-index of SP) is inferred from the aggregated values (Table 2).

Table 1: Fuzzy input variables

Input variable	Linguistic variables
Consumer's satisfaction -index	Very low
	Low
	Medium
	High
	Very high
Broker's satisfaction -index	Very low
	Low
	Medium
	High
	Very high
Normalized cost	Very low
	Low
	Medium
	High
	Very high

Table 2: Fuzzy output variable

Output variable	Linguistic variables
Trust-index of SP	Very low
	Low
	Medium
	High
	Very high

SIMULATION

We simulated the architecture with 10 domains, 40 to 80 brokers and 400 consumers, 400 SPs with 40 entities acting as both consumer and SP. For the purpose of simplicity ten different service types with fixed criticality rates were considered; however, this can be easily extended without loosing generality.

Given the total number of requests to be generated and the mean of the distribution, the generation of number of requests to be issued during the ith time instant is according to a Poisson distribution. Here, we used 10000 requests with the mean value of 30. The probability mass function for the Poisson process is

$$P\ (X = r/T = t) = (e^{(\lambda i)}) * (\lambda t)^r)/r!$$

Which gives the probability that all events will occur during a duration of t. The maximum probable number of requests to be generated at the time instance i (X) may be found by computing the total number of probable requests for a time duration of i (R_i) and subtracting that from the accumulated sum of number of requests generated till the last instance ($\sum R_j$, j = 1 to i-1). Thus, $X = R_i - \sum R_j$ where j = 1 to i-1, R_i = r for which P(X = r, T = i) is maximum. The concise flow of simulation is given below.

Broker module: This module simulates the work of a broker.

While (true)

The broker has two threads running simultaneously:

- One thread waits to handle requests from the SP or the consumer
- If the request is from an SP for registration under the broker then action is performed accordance with logic explained in the study.
- If the request is from an SP for deregistration then action is performed accordance with logic explained.
- Else If the request is from a consumer to request for a service.
    - The process is done as given in the study.
    - It then accepts the feedback from the consumer and stores it in its database.

- The other thread is used for periodic updation of the SP's trust-index
    - Process is done according to the logic explained in the study.
    - After recalculation of trust-index of SP, the trust-index value is forwarded to other brokers to which the SP is associated with.

Consumer module: While (true)

The consumer has one thread (for each consumer) running

- It sends a request to register itself to a broker which is done according to logic explained in the study.
- After registration, it sends its service requests to the broker and the request is serviced in the study.
- After using the SP's service, it sends the feedback about the SP's service.
- If the broker is unable to find an appropriate SP, the consumer is free to forward its request to another broker.

SP Module: While (true)

The SP has one thread

- It sends a request to register itself to a broker which is done according to logic explained in the study.
- After registration, it waits continually to accept the requests from consumers and sends the responses to the consumers.
- After each transaction with the consumer it sends a feedback for that consumer to the broker.
- It sends a request to deregister itself from a broker which is done according to logic explained in in the study.

RESULTS AND DISCUSSION

The normalized cost-loss of the consumer is plotted against the number of transactions in Fig. 2. This cost-loss has been compared for two grid scenarios, the scenario with the presence of the broker (with trust) and without the presence of the broker (without trust). By comparing the two scenarios, it can be concluded that the presence of the broker (with trust) provides a lesser cost-loss to the consumer.

The number of messages is plotted against the number of transactions is shown in Fig. 3. A comparison between the same two scenarios mentioned above has been made. It has been found that the number of messages for with broker scenario is far higher than the broker-less scenario. But with the introduction of the periodic updation SP trust-index, the number of messages drastically reduce even in a with broker scenario.

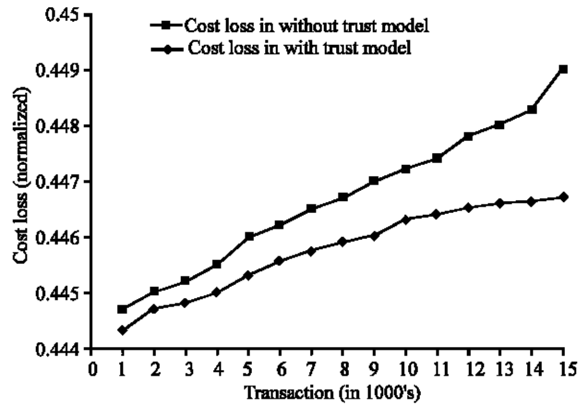

Fig. 2: Comparison of cost-loss with and without trust

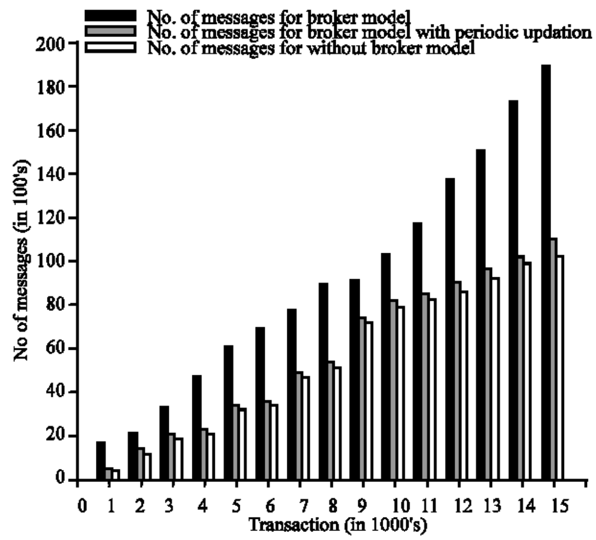

Fig. 3: Comparison of number of messages with and without broker

The broker trust-index in the presence of vicious SPs has been plotted against the number of transactions in Fig. 4. It is seen that with the deregistration concept the broker's trust-index is almost maintained since the untrustworthy SP's are deregistered.

The accuracy of trust-index of SP is plotted against the number of transactions is shown in Fig. 5. As explained in the study, if the trust index of SP is updated periodically based on current trust index of SP (not after every transaction), it is shown that there is a compromise of the accuracy of trust-index of the SP.

Figure 6 shows the variance in the accuracy of trust-index of SP between periodic-updation and updation after every transaction experienced by any two SP's.It is seen that it is nearly the same and hence the accuracy of the overall system is not affected.

The impact of maliciousness from the SP by providing inaccurate results, results in poor feedback from

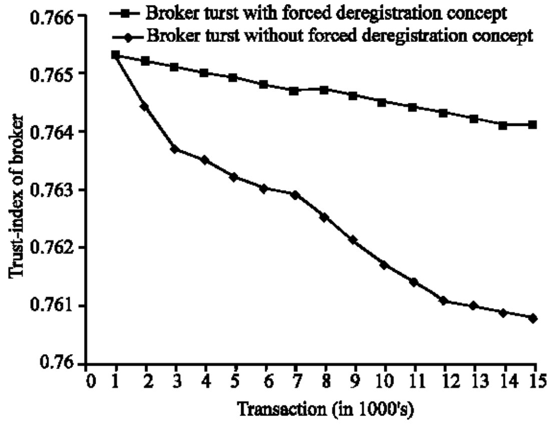

Fig 4: Comparison of Broker trust-index in with broker model, inclusive of forced deregistration of SP and non-inclusive of forced deregistration of SP

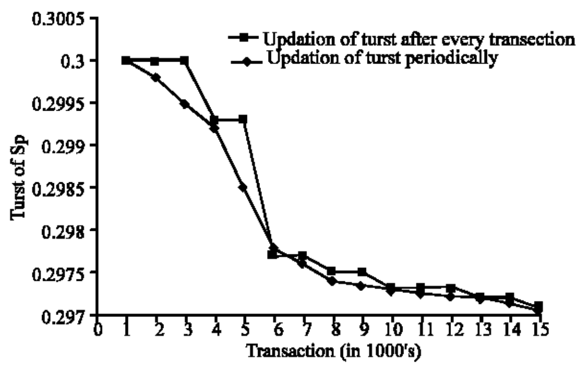

Fig. 5: Comparison of accuracy of trust-index with and without updation of trust-index after every transaction

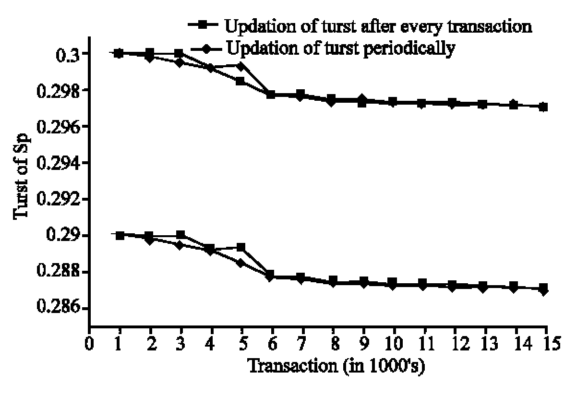

Fig. 6: Comparison of the variance in the calculation trust-index of two SP's with and without updation after every transaction

the consumer. The impact of the consumer and broker feedbacks on the trust-index of SP has been plotted in

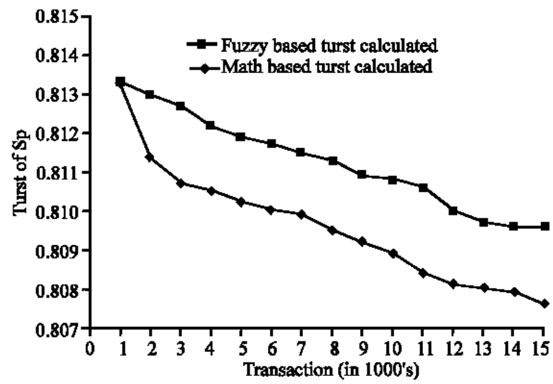

Fig. 7: Comparison of fuzzy based and math based model for trust calculation

Fig. 7 against the number of transactions using both the mathematical and a fuzzy-based model. The inputs considered for fuzzification process are consumer-satisfaction, broker-satisfaction and normalized cost of the transaction. The SP with better trust-index is gradually reduced in fuzzy-based model in comparison to mathematical model when there is continuous malicious behavior of SP in order not to punish the SPs immediately.

CONCLUSION

In this study, we proposed an integrated reputation and policy-based broker architecture to improve the performance of the system without any runtime problems. The broker publishing concepts (SOA) ensures that every new broker entity that enters a domain is published to provide an opportunity for SPs to associate themselves with it. This study gives SP's the freedom to SP's to associate themselves to any broker.The disassociation of an SP from the broker had been dealt with in two specific scenarios. This study provides brokers the freedom to allow or disallow any SP to be associated with it based on SP's trust-index. This study deals with assigning an initial trust-index to the SP using its presence-index instead of random assignment used in other trust models (Azzedin and Meheswaran, 2002, 2004; Varalakshmi *et al.*, 2006). This model also gives consumers the freedom to assign priorities to Qos parameters while requesting for a service The periodic updation of trust-index of SPs compared to updation after ever transaction reduces the workload of the broker and the number of message exchanges without affecting the accuracy of the system to a greater extent.

REFERENCES

Azzedin, F. and M. Maheswaran, 2002. Evolving and managing trust-index in grid computing systems, Proceeding of the IEEE Canadian Conference on Electtrical Computer Engineering.

Azzedin, F. and M. Maheswaran, 2002. Integrating Trust-index into Grid Resource Management Systems, The International Association for Computers and Communications, Vancouver, B.C., Canada: IEEE Computer Society Press.

Azzedin, F. and M. Maheswaran, 2004. A Trust-index Brokering System and Its Application to Resource Management in Public Resource Grid, International Parallel and distributed Computing Symposium IPDPS.

Del Vecchio, D., M. Humphrey, J. Basney and N. Nagaratnam, 2005. CredEx: user-centric credential management for grid and Web services, IEEE International Conference on Volume, Issue.

Erika Martinez Ramirez and V. Rene Mayorga, 2007. A Cascaded Fuzzy Inference System for Dynamic Online Portals Customization, International Journal of Intelligent Technology, Vol. 2.

Ionut Constandache, Daniel Olmedilla and Wolfgang Nejdl, 2005. Policy Based Dynamic Negotiation for Grid Services Authorization, REWERSE-RP.

Piero Bonatti, Claudiu Duma, Daniel Omedilla and Nashhid Shamehri, 2005. An Integration of Reputation-based and Policy-based Trust-index Management, REWERSE-RP.

Samia Nefti, Farid Meziane and Khairudin Kasiran, 2005. A Fuzzy Trust-index Model for E-Commerce, Proceedings of the seventh International Conference on E-Commerce Technology (CEC).

Shanshan Song, Kai Hwang and Runfang Zhou and Yu-Kwong Kwok, 2005. Trust-indexed P2P Transactions with Fuzzy Reputation Aggregation", IEEE Internet Computing.

Varalakshmi, P., S.Thamarai Selvi and K. Narmatha, 2006. A broker Architecture Framework for Reputation-based Trust-index Management in Grid, Proceeding of International Conference on Information Security, ICIS.

Varalakshmi, P., S. Thamarai Selvi and M. Pradeep, 2007. A multi-broker trust management framework for resource selection in grid, IEEE Workshop IAMCOMM.