ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org
Vol 5, Issuse.4 Oct2017

Evolving and Managing Trust in Grid Computing Systems*

Gayatri, Dr. Prabhu , Imtiyaz Khan

Abstract

Grid computing refers to a system in which computing tasks are dispersed over several computers in a network, each of which operates independently but shares its resources with the others. A fundamental objective of a Grid environment is to promote interactions across domains and boost trust among domains to use or share resources (a) without giving up control of their own resources and (b) while protecting the privacy of others. In order to make such geographically dispersed systems more appealing and dependable for day-to-day usage, it is necessary to address the concept of "trust." In this work, we propose a two-stage approach to establishing trust: (a) confirming an entity's identification and the scope of its allowed actions; and (b) monitoring and controlling the entity's behavior and establishing a trust level based on that activity. Many studies have focused on the confidence in one's identity, while the trust in one's actions has received less attention. We provide a formally defined concept of behavior trust and reputation and talk about a behavior trust management architecture that mimics the evolution and administration of behavior trust in Grid computing systems.

Keywords: safety, confidence, and Grid computing

1. Introduction

Recent years have seen an uptick in study of grid computing systems [FoK99, FoK01], which provide a digital infrastructure for the managed distribution of resources across organizational boundaries. In a globally dispersed setting, an organization may tap into resources it wouldn't have access to on its own. Some organizations are wary of adopting a virtual architecture like the Grid due to the perceived security risks involved with "sharing" infrastructure or operations. Due to the importance and fragility of the data or information at hand, such organizations would rather employ their own "closed box" infrastructure. It's not only wasteful overall, but also expensive, for each particular organization.

Grid computing would be more enticing if there were trust zones where entities could consume resources or develop services without risk. Many scholars [MeO01, AdF99, AbH00, DaD01] have attempted to tackle the multifaceted topic of trust from a variety of angles. Trust is broken down into two subcategories: trust in one's identity and trust in one's actions. Encryption, data concealing, digital signatures, authentication protocols, and other access control technologies all play a role in identity trust because they allow for the verification of an entity's authenticity and the determination of the authorizations to which the entity is entitled. While the focus of behavior trust is on a more generalized concept of reliability. For instance, it is not clear from a digitally signed certificate whether or not the issuer is an industrial spy [AbH00], and it is not clear from digitally signed code whether or not it was produced by skilled programmers.

In this study, we suggest a trust model for Grid computing systems, one that takes into account the nature of trust, how it develops in response to interactions between entities, and how it is maintained over time. Unless otherwise specified, for the remainder of the article, the word "trust" will refer to behavioral trust. In Section 2, we'll define trust and reputation and go through some basic mechanisms for calculating them. In Section 3, we provide a holistic trust model for Grid systems. In Section 4, we examine the concept of trustworthy domains in a Grid setting. Section 5 illustrates the creation and maintenance of a trust relationship using an example trust transaction involving two domains. In Section 6, we briefly describe previous research in this area.

Asst. Professor
Department of CSE
gayatri12islclg@gmail.com, drprabhu42@gmail.com, imtiyaz.khan.7@gmail.com
ISL Engineering College.
International Airport Road, Bandlaguda, Chandrayangutta Hyderabad - 500005 Telangana, India.

2. Trust and Reputation

2.1. Definition of Trust and Reputation

Trust is a multifaceted concept that involves having faith in the sincerity, competence, and dependability of another person or organization. There is disagreement in the literature on what constitutes trust management and what exactly constitutes trust [Mis96, GrS00, AbH00]. This study will use the following definition of trust:

Trust is the conviction that another person or thing can and will carry out an assigned task or accomplish an intended goal, when this conviction is conditional on the other person's or thing's subsequent actions and is applicable only in a certain setting and at a particular moment.

In other words, the intensity of one's conviction might vary from high to low, or from trustworthy to untrustworthy. This level of trust (TL) is established via previous interactions....and intended for use in a particular setting. An organization y may allow an organization x to access its data storage facilities, but not its data processing facilities. Because the TL between two entities today is not necessarily the same as it was a year ago, the TL is indicated within a specific period.

Entities may depend on others to provide accurate information when making choices based on trust. To provide an example, an unknown machine Mj's reputation may help an unknown entity x decide whether or not to There are a number of factors that need to be taken into account when calculating trust and reputation. One, over time, trust naturally wanes. If x had a high degree of trust in y five years ago based on their history together, that level of trust would likely have decreased unless they had some kind of interaction in the meantime. Time-dependent degradation is also seen in

reputation is at stake Second, organizations may and do establish alliances, and those organizations are more likely to place confidence in their friends and commercial partners than they are in other organizations. Finally, x's confidence in y is founded on both x's personal experience with y and y's reputation; hence, the trust model must be able to calculate the ultimate trust based on a mix of direct trust and reputation while giving each factor its own weight.

Two classes of entities, Di and Dj, are distinguished. To calculate the trust relationship between the two domains in a given context c at a given time t, we need to know not only the direct relationship between Di and Dj in that context c at that time t, expressed as (Di, Dj, t, c), but also the reputation of Dj in that context c at that time t, expressed as (Di, Dj, t, c) (Dj, t, c). A direct relationship's weight is, whereas a reputational connection's is. For Di, Dj's "trustworthiness" is determined less by Dj's reputation than by her personal connection to Dj. Consequently, is more than. The decay function (Y(t tij, c)) is multiplied with the trust level in the direct-trust table (DTT), where c is the particular context for the trust connection, t is the utilize Mj. In this study, we shall adopt the following definition of reputation:

A thing's reputation is what other things think it will do in a particular situation based on what they know about its conduct in similar situations in the past.

2.2. Computing Trust and Reputation

In all domains k, the recommender's trust factor is denoted as R(Dk, Dj). To put it another way, in real-world systems, RTT and DTT will be identical, since entities would utilize the same data for both purposes. We established the recommender trust factor R to avoid cheating by collusions among a group of domains due to the fact that a domain's reputation is mostly reliant on what other domains say about it. Therefore, R takes on a value between 0 and 1; it will be greater if Dk and Dj are unrelated strangers, and it will be less if Dk and Dj are friends or business associates.

$$\Gamma(D_i, D_j, t, c) = \alpha \times \Theta(D_i, D_j, t, c) + \beta \times \Omega(D_j, t, c)$$

$$\Theta(D_i, D_j, t, c) = DTT(D_i, D_j, c) \times \Upsilon(t - t_{ij}, c)$$

$$\Omega(D_j, t, c) = \sum_{n} RTT(D_k, D_j, c) \times R(D_k, D_j) \times \Upsilon(t - t_{kj}, c)$$

3. Trust Model

3.1. Overview

$n\ k=1$

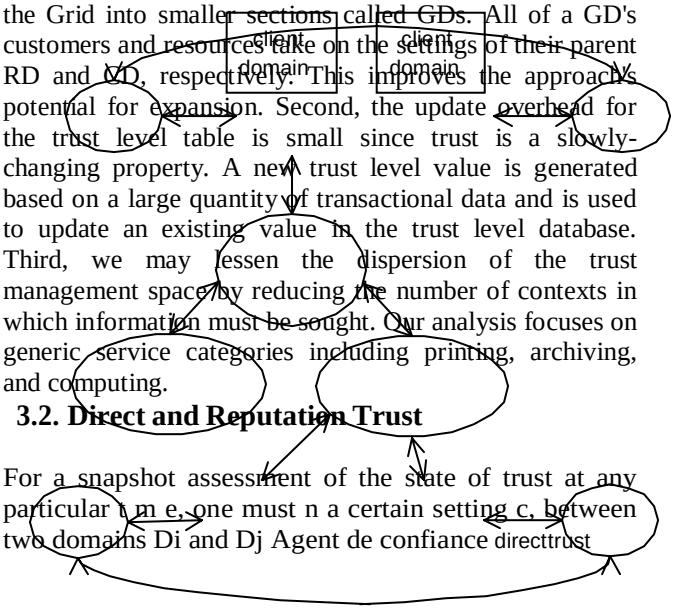
(D_k)

current time, and tij is the time of the last update or the last transact- ion between Di and Dj. Information that was well-received from an entity five years ago can be ill-received now depending on the authenticity of the information and how trustworthy the entity is today, making the time component t as described before highly essential. Dj's reputation is found by averaging the product of the trust in the RTT, the decay function ((t tkj, c), and the time since Dj's last positive reputation update (tkj, c).

Figure 1 depicts the Grid's general trust model, which identifies the Grid's subdivided "Grid domains" (GDs). We assign two virtual domains, a resource domain (RC) representing the GD's resources and a client domain (CD) representing the GD's clients, to every GD we create. Trust agents are implemented in each GD with the capabilities to (a) update the trust tables of the GDs, (b) let entities to join GDs and inherit their trust characteristics, and (c) apply a decay function to represent the loss of trust across domains.

Creating and updating the trust level table in a naive manner may be wasteful in a system of the Grid's

magnitude. Our approach employs a number of techniques to make this process more effective. To begin, we partition the Grid into smaller sections called GDs. All of a GD's customers and resources take on the settings of their parent RD and VD, respectively. This improves the approach's potential for expansion. Second, the update overhead for the trust level table is small since trust is a slowly-changing property. A new trust level value is generated based on a large quantity of transactional data and is used to update an existing value in the trust level database. Third, we may lessen the dispersion of the trust management space by reducing the number of contexts in which information must be sought. Our analysis focuses on generic service categories including printing, archiving, and computing.



3.2. Direct and Reputation Trust

For a snapshot assessment of the state of trust at any particular t m e, one must n a certain setting c, between two domains Di and Dj Agent de confiance

Context	Domains		
	D_1	...	D_j
c_1	$TL^{c_1}_{.k1}$	...	$TL^{c_1}_{.kj}$
.	.	.	.
.	.	.	.
c_i	$TL^{c_i}_{k1}$	...	$TL^{c_i}_{kj}$

Direct and Reputation Trust Weights

Table 1. Description of the required trust levels.

Trust Level (TL)	Description
A	very low trust level
B	low trust level
C	medium trust level
D	high trust level
E	very high trust level
F	extremely high trust level

3.3. There are two parts to this puzzle that must be solved: (a) the personal connection (direct trust) and (b) the reputational link (indi- rect trust based on recommendations). The DTTs will be managed by the respective domain trust agents as stated in Table 2. This table shows that there is a direct connection between Dk and Dj for a given ci since Dk may make use of re- sources or install services utilizing Dj's resources. Due to the asymmetric nature of a direct trust relationship, the quality of this connection will be evaluated differently by each of the two parties involved (see Table 1). Di may depend on suggestions from other domains while interacting with Dj, in addition to the direct trust connection (i.e., asking for the reputation of Dj). For this reason, as shown in Sections 3.5 and 3.6, the trust agent for each domain will assess both the direct and recommender trusts.

Table 2. Direct trust table maintained by D_k.

3.4. (= 0) with recommenders with whom it has no personal connection.

3.5. Decay Function

3.6. Eventually, trust will fade, just like any other connection. For example, unless Di and Dj have reestablished contact during the last five years, the strength of their TL is likely to have diminished. To account for this decline when modeling trust across domains, we included a decay function into our trust model. By observing how much time has passed since the previous exchange between Di and Dj, we may calculate the decay function Y(t tij, c). It's possible that the TL decay rate and the variables that speed it up or slow it down are different in each domain. Di's familiarity with Dj's domain, for instance, means that both share the same set of legal obligations (i.e. from the same union, country, etc.). Therefore, Di may choose to accelerate the TL decay for domains in unfamiliar surroundings relative to domains in familiar environments.

3.7. Trust Inheritance

Entities in such a decentralized setting may enter or leave a domain Di at any moment. That's why it's important that such an environment's trust model include strategies for dealing with trust in organizations. In the following ways, our trust model accounts for this reality. Whenever an entity x becomes a member of a domain, it automatically receives all of the TLs from both the DTT and the RTT of the domain. However, x's lack of long-term Di service may cause some skepticism from Di's other domains. Therefore, each entity is assigned a member weight that indicates whether the entity is a new, recent, or old member with its domain. It is up to the specific domain to determine what makes an entity to

fall into one of these member weight categories.

3.8. Evolving Trust

Domains may use our methodology to construct their TLs from scratch without expert guidance or vetted recommendations. It may be argued that, as a novice, you are always at risk of being taken advantage of by a rogue site that seems to give "help" but really has evil intentions. It's true that there's a lot of mystery around what exactly other people do when you're a beginner to a field. Our trust paradigm, however, is meant to shield newcomers from harm. Pretend for a moment that Di is a newbie who is eager to speak with Dj. A RTL will be present in each of these areas. Since Di is a new member of the network, it may increase protection for its data and programs by setting its RTL to F, which indicates that no preexisting trust connection exists between the two parties. Di can develop its own trust values by its interactions with other domains.

Table 3. Recommendations received by D_i

Context	Domains	
	D_1	D_2
printing service	*D*	*C*

4. Trusted Domains

Integrating "trust" into networked computing Systems with built-in trust awareness allow for the complete separation of client and server groups. into known and secure online spaces. Trusted domains in a Grid-like distributed computing environment boost and stimulate the use of business-to-business or organization-to-organization applications, which in turn: (a) generate additional application-to-service mappings and (b) may give rise to novel types of service models. As a result of this reduced security burden, both application performance and resource usage will increase..

5. Trust Transaction Example

As an example of how our concept may be used in the real world, we'll look at a scenario in which Domain I trusts Domain II to print documents for me, Domain II trusts Domain I to print documents for me, and Domain I trusts Domain II to print documents for me. Assume for the moment that Di is a new player and has no data yet in either the DTT or RTT. Another.dj site is on the hunt for a "printing provider" to help it publish its annual report. While issues about "trust" exist for both Di and Dj, we're more interested in how Di's "trust" in Dj develops and grows as a result of this interaction. As shown in Figure 1, a resource management agent contacts Di as a potential RD since it delivers the requested service. Since Di is a novice and does not yet have a mutually beneficial trust relationship with Dj, Di defaults its RTL to the forward direction. Di may also depend on recommendations, as it has been told the two things about Dj shown in Table 3.

Di evaluates the direct trust connection with Dj (i.e., Di updates its DTT) by checking if Dj follows its RTL after the transaction between Di and Dj has begun. Di performs this assessment by two methods: (a) determining whether Dj is an abusive do- main through an audit trail analysis [Lun93] by identifying unsuccessful instructions sent by Dj, and (b) monitoring sequences of system calls to detect an abnormal behavior of Dj [HoF98]. Assume that Di uses the classifications in Table 4 to categorize the actions of other domains. In addition, let's imagine that Di does identify Dj's aberrant conduct and gives a trust value of 3, which would translate to a TL of D.

Di's DTT may be revised since TL(tij, c) = 3, as described in Section 3.8. In the beginning, DTT (Di, Dj, c) was 0, but now it is: DTT (Di, Dj, c) = TL (tij, c). As a result, Di may establish a brand-new direct trust connection (i.e., update its DTT), with D as the new TL for the resulting DTT (Di, Dj, c).

In addition, Di is now able to:

Table 4. D_i' s classification system

Classification range	Classification description	Trust level assigned
0 − 2	very little harm	*E*
2 − 4	little harm	*D*
4 − 6	medium harm	*C*
6 − 8	high harm	*B*
8 − 10	very high harm	*A*

the RTT in a manner consistent with that described in Section 3.8. In realistic settings, RTT and DTT will be equivalent since entities will utilize the same data to assess direct linkages and make recommendations. Thus, we have a value of D for RTT (Di, Dj, c).

Third, Di must update its recommender trust factor table (i.e., update R) in order to assess the recommender's reliability, as described in Section 3.6. D1 and D2 both said that Dj should be trusted, although they made different recommendations (D and C, respectively). As a result of Di's conversation with Dj, Di has learned that D is Di's TL. Thus, the R factors for D1 and D2 are set at 1 and 0.6, respectively. Based on these considerations, it seems that D1's recommendation of Dj was more accurate than D2's.

6. Related Work

Identity trust is the focus of many trust models and trust management systems, including the Pretty Good Privacy (GnuPG) Public Key Infrastructure (PGP) [MeO01] and the X.509 [AdF99] standards. However, these approaches to trust do not include any methods to keep tabs on how people's confidence in one another evolves over time. In addition, neither the trust models nor the trust management systems take into account the fact that entities need to learn from their experiences in order to dynamically update their trust levels [GrS00].

In [AbH00], the authors suggest a paradigm for encouraging behavioural trust based on experience and reputation. Entities may use this trust-based paradigm to choose which other entities they can put their faith in, and they can also adjust how well they understand the suggestions made by other entities.

As part of this study, a policy specification language named Ponder [DaD01] was created to facilitate behavior trust, and a survey of trust in Internet applications is published in [GrS00]. Policies for authorisation and security management may be developed with the help of Ponder. The scope of Ponder is being broadened to include trust relationships between entities that span different organizational domains, which may be rather abstract and intricate.

To a great extent, our model builds upon [AbH00, DaD01] While it's true that (a) trust degrades with time, (b) an organization may place greater faith in its closest friends and partners, and (c) our trust model employs a process whereby trust values re-

Check out the Publish Rankings result from personal connections are given greater weight than those resulting from an organization's reputation, and (d) we allow for inheritance in our trust model.

7. Conclusions

The Grid is being promoted as a computational infrastructure that will allow shared resource pools to be used by researchers from different institutions. Privacy, secrecy, and individual agency are only a few of the issues that have been raised in response to the concept of "sharing." That's why "trust" is an issue that needs fixing in a networked setup. From our perspective, there are two parts to building trust: (a) validating the identification of an organization and the actions it is permitted to do, and (b) monitoring and managing the entity's actions after that identity has been established. Encryption, data concealing, digital signatures, and access control are only some of the methods that have helped improve people's ability to trust each other's identities. In this study, we present a framework for managing trust that may foster and sustain relationships of trust based on both direct and reputational criteria. To demonstrate how our approach adapts and controls trust across two domains, we give a sample application.

References

Supporting Trust in Virtual Communities, A. Abdul-Rahman and S. Hailes, Hawaii International Conference on System Sciences, 2000 (AbH00).

Certificate Management Protocols for the Internet X.509 Public Key Infrastructure, RFC 2510, C. Adams and S. Farral, 1999.

"Decen- tralized trust management," IEEE Conference on Security and Privacy, 1996, M. Blaze, J. Feigenbaum, and J. Lacy.

Utilizing the KeyNote Trust Management System, by M. Blaze, AT&T Research Laboratories, 1999 (Bla99).

Nikos Damianou, Nikos Dulay, Emilia Lupu, and Michael Sloman, "The Ponder policy specification lan- guage," Workshop on Policies for Distributed Systems and Networks, 2001 [DaD01].

The Grid's internal structure is described in detail in an article published in 2001 in the International Journal of Supercomputing Applications by I. Foster, C. Kesselman, and S. Tuecke (cited as FoK01).

[FoK99] I. Foster and C. Kesselman (eds.), The Grid: Blueprint for a New Computing Infrastructure, The year is 1999, and the place is San Francisco, California, and the publisher is Morgan Kaufmann.

[GrS00] T. Grandison and M. Sloman, “A study of trust in Internet applications,” IEEE Communications Surveys & Tutorials, Vol. 3, No. 4, 2000.

European Symposium on Research in Computer Security (ESORIC'92), 1992; N. Habra, B. L. Chalier, A. Mounji, and I. Math- ieu, "ASAX: Software architecture and rule- based language for universal audit trail analysis."

Journal of Computer Security, Volume 6, Issue 1, Pages 151-180; S. A. Hofmeyr, A. Somayaji, and S. Forrest, "Intrusion detection utilizing sequences of system calls."

T. F. Lunt, "Detecting intruders in computer sys- tems," Conference on auditing and computer technology, 1993 [Lun93]].

Handbook of Applied Cryptography, Fifth Edition, A. J. Menezes, Peter C. Oorshot, and Stuart A. Vanstone, CRC Press, New York, 2001 [MeO01].

"Trust in Modern Societies" by B. Misztal, Polity Press, Cambridge, MA, 1996.

Journal of Computer Security, Volume 10 Issue 1 1994, Pages 39–49, S. E. Smaha and J. Winslow, "Misuse detection tools."