Journal of Network and Computer Applications 36 (2013) 274–283

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

RETENTION: A reactive trust-based mechanism to detect and punish malicious nodes in ad hoc grid environments

Reinaldo B. Braga [a,*], Igor A. Chaves [b], Carina T. de Oliveira [a], Rossana M.C. Andrade [b], José Neuman de Souza [b], Hervé Martin [a], Bruno Schulze [c]

[a] Joseph Fourier University (UJF) – University of Grenoble 681, Rue de la passerelle, Saint Martin d'Heres, Grenoble, France
[b] Federal University of Ceará (UFC) Av. Mister Hull, s/n – Campus do Pici – Bloco 942-A, Fortaleza, CE, Brazil
[c] National Laboratory for Scientific Computing (LNCC) Av. Getulio Vargas, 333, Petrópolis, RJ, Brazil

ARTICLE INFO

Article history:
Received 8 September 2011
Received in revised form
24 March 2012
Accepted 3 June 2012
Available online 15 June 2012

Keywords:
Security
Grid computing
Intrusion detection
Ad hoc grid environments

ABSTRACT

In ad hoc grid environments, resources are not always available since nodes can spontaneously connect and disconnect at any time. Thus, these environments demand the correct execution of tasks to guarantee good performance. However, there are malicious users that affect the normal operation of these grids. These users modify tasks results and even cheat security mechanisms. Therefore, to assure high performance in these grid computing scenarios, it is essential to use punishment procedures based on trust models. These solutions have been used in wireless ad hoc networks, but not in the context of ad hoc grid computing. Thus, in this paper, we first present an analysis of mathematical trust models in ad hoc grid scenarios, using different ways to treat detection information passed on by other nodes. Then, we provide a comparison and a performance evaluation of these models using a grid simulator platform. Besides that, we choose the most accurate trust model among the evaluated ones to propose RETENTION: a reactive trust-based mechanism to detect and punish malicious nodes in ad hoc grid environments. Simulation results demonstrate the effectiveness of the proposed approach in detecting and punishing up to 100% of malicious nodes without generating false-positives. The results can be a valuable tool for network designers in planning trust models in ad hoc grid network deployments.

1. Introduction

Grid computing is a term referring to environments composed of heterogeneous resources geographically distributed that allow the creation of unified systems with high computing and storage performance (Foster and Kesselman, 2003). It provides resources for a large number of applications that go from simple shared disk storage systems to complex data analysis and virtualization.

Grid environments are classified according to the availability of resources. When the grid composed of permanent resources, it is called a dedicated grid. For instance, the user of a dedicated grid can rely on the availability of resources to execute his tasks. Conversely, a grid with dynamic resources is called an ad hoc grid and the resources are not always available because nodes can spontaneously connect and disconnect any time (Dornemann et al., 2008). This work focuses on ad hoc grid computing environments.

Likewise to wireless ad hoc networks (Jurdak et al., 2004), ad hoc grid computing does not have a predefined infrastructure. It relies on the cooperative behavior of nodes to work properly. Thus, nodes that share their resources must behave in accordance with the specifications determined for the grid environment and they have to process all the tasks submitted to their shared computational resources. Nevertheless, the assumption that nodes behave correctly may be false, due to malicious behavior. For instance, malicious nodes may change the result of tasks and, thus, affect grid performance. Therefore, a mechanism to detect malicious nodes in the grid is fundamental. Furthermore, once malicious nodes are detected, a mechanism to punish these nodes is necessary.

Several approaches in literature tackle this problem through traditional security mechanisms. We can broadly classify these approaches into two categories of security mechanisms: preventive and reactive. Preventive mechanisms, as the name suggests, are used as the first wall against attacks in computational environments. In other words, preventive mechanisms are applied to avoid an attack in the environment (Turkmen and Crispo, 2008; Smith et al., 2009; Kanda et al., 2010). Reactive mechanisms, on the other hand, detect nodes that are able to circumvent the preventive mechanisms and perform their malicious behavior. In this paper we focus on reactive mechanisms.

* Corresponding author. Tel.: +55 85 3366 9797.
E-mail addresses: braga@imag.fr, reinaldobraga@gmail.com (R.B. Braga), igor@great.ufc.br (I.A. Chaves), oliveira@imag.fr (C.T. de Oliveira), rossana@ufc.br (R.M.C. Andrade), neuman@ufc.br (J. Neuman de Souza), herve.martin@imag.fr (H. Martin), schulze@lncc.br (B. Schulze).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.06.002

The Intrusion Detection System (IDS) (Huang and Lee, 2003) is the most common reactive mechanism. It uses the information about normal or anomalous behaviors to detect and punish malicious nodes in the grid. The punishment can be performed following two strategies: local and distributed. Local punishments are based on detections carried out by the own user, and detection alerts provided by other users in the environment are not considered. Nevertheless, distributed punishments make use of all detection alerts provided by the other users to perform a punishment.

It is important to create an accurate distributed punishment system as local techniques punish malicious nodes separately without informing punishments to other nodes (Sanchez et al., 2007). Therefore, each node exchanges information about detections performed locally and can initiate a distributed punishment procedure. Note that, in order to create more accurate distributed punishment systems, it is necessary to use a trust mechanism, because the nodes that send information to the grid have to be reliable.

Although trust models have been mentioned in the past for wireless ad hoc networks, we believe that this work represents the first comprehensive study of these trust models in the context of ad hoc grid computing. Thus, this paper first describes and evaluates trust models using a grid simulator platform in order to point out the best mechanism to detect malicious nodes in ad hoc grid environments. In particular, we show that some models are able to reduce the amount of false-negatives without generating false-positives. Based on this analysis, we then propose RETENTION: a reactive rrust-based mechanism to detect and punish malicious nodes in ad hoc grid environments. Simulations show that our proposed mechanism efficiently detects and punishes malicious nodes that corrupt tasks in the grid.

The rest of this paper is structured as follows: Section 2 discusses related work. In Section 3, we introduce the characteristics of trust models in ad hoc grid computing and present the performance evaluation of these trust models. Section 4 presents our proposal. Section 5 introduces the simulation environment and the assumptions for the experiments. The simulation results are presented in Section 6. Finally, Section 7 summarizes our contributions and identifies avenues for future research.

2. Related work

In this section we present an overview of research efforts devoted to improve the performance of grid environments. First, we present framework authorization solutions for grid computing. Then, we discuss the importance of intrusion detection systems and trust solutions in ad hoc grid environments.

2.1. Authorization frameworks

One of the first security solutions for grid computing was presented by the Globus Alliance (The Globus Alliance, 2011) and it is known as Community Authorization Service (CAS) (Pearlman et al., 2002). CAS is a framework that uses an authentication tool through a central administration entity called CAS server, which is controlled by a community administrator with coarse-grained authorization privileges. The CAS server defines rules to manage fine-grained authorization permissions among the grid users. For example, a rule can be a node, which provides its resources in full time, could have a high level of trust because other grid nodes take into account its resource is trustworthy. Analyzing the CAS, we can observe that this framework is efficient in dedicated grids because we can know all entities in the grid. However, the CAS is not an ideal tool to be used in ad hoc grid computing, because an entity that shares its resources could not be a trusted entity. Besides that, the requirements for a central administration (the CAS server) and a community administrator are not feasible in ad hoc grids.

The Virtual Organization Membership Service (VOMS) (Alfieri et al., 2003) is another example of community-based authorization framework in which every Virtual Organization (VO) has an associated VOMS server and a VOMS administrator. In VOMS-based system however, grid users define a set of local policies that are distributed to the community users via the VOMS server. Although VOMS differs from CAS framework in its representation of the community privileges, it still relies on a pre-established community administrator (the VOMS server). Its dependence on a centralized attribute authority prohibits VOMS-based systems from supporting the structure and control-independent requirements of ad hoc grids environments (Zhang et al., 2008).

Akenti (Thompson et al., 1999) is a certificate-based access control system that provides distributed policy management mechanisms. The access control on resources is based on policies expressed by multiple authoritative entities called stakeholders. First, the policies are specified in terms of what attributes a user must possess in order to perform a specific request. Note that each stakeholder is allowed to place policies on who and how the service can be used. Then, the policies are conveyed in certificates and dispersed over the network (e.g. LDAP server and web server). Finally, the policies from all authoritative stakeholders are combined to decide if a request may be performed or not. The Akenti system is dependent on Public Key Infrastructure (PKI) (Thompson et al., 2003), which assumes the existence of a pre-established infrastructure. Therefore, this approach is not suitable to ad hoc grids.

Following a similar approach, Lorch et al. (2004) proposed a privilege management and authorization (PRIMA) system that supports the creation of multiple administrative authorization entities for service utilization. In other words, PRIMA allows multiple users to delegate access to services for which they are authoritative, based on the aggregate set of *privilege attributes* (e.g. file access permissions, network access, user quotas) provided by the requesting user (Lorch and Kafura, 2004). With this model, Amin et al. (2005) proposed the Ad Hoc Grid Security Infrastructure (AGSI), which aims to use the relation between certification authorities of the ad hoc grid, decentralizing the grid's authentication mechanism. Thus, the AGSI unites a distributed mechanism of authentication and of user access control to the ad hoc grid services.

Relevant solutions such as PERMIS (Chadwick and Otenko, 2002), Cardea (Lepro, 2003), XPOLA (Fang et al., 2005), Shibboleth (Gao and Tan, 2011), MyProxy (The MyProxy Project, 2012), and GridShib (The GridShib Project, 2012) are also designed to support grid authentication model. Most of these approaches assume that the requesting user has some preliminary knowledge about the authorization entity (Jie et al., 2010). However, these approaches are not expressive and flexible enough to deal with dynamic environments like ad hoc grid (Barton et al., 2006). For example, a malicious node is still able to circumvent these preventive security mechanisms and perform its attack successfully. This way, it is important to use a reactive security mechanism able to detect a malicious node and send alerts about these detections in the grid, to make the system carry out distributed punishment with greater precision.

2.2. Intrusion detection system (IDS)

The distributed nature of ad hoc grid computing environment makes it vulnerable to malicious attacks. For this reason, additional security measures are mandatory to identify typical attack behavior. Intrusion Detection System (IDS) is a reactive mechanism that generates a security alert for each intrusion event