Expert Systems with Applications 41 (2014) 3831–3849

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Behavioral modeling and formal verification of a resource discovery approach in Grid computing

Alireza Souri, Nima Jafari Navimipour *

Department of Computer Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran

ARTICLE INFO

Keywords:
Grid computing
Formal verification
Behavioral modeling
Resource discovery
Model checking
NuSMV

ABSTRACT

Grid computing is the federation of resources from multiple locations to facilitate resource sharing and problem solving over the Internet. The challenge of finding services or resources in Grid environments has recently been the subject of many papers and researches. These researches and papers evaluate their approaches only by simulation and experiments. Therefore, it is possible that some part of the state space of the problem is not analyzed and checked well. To overcome this defect, model checking as an automatic technique for the verification of the systems is a suitable solution. In this paper, an adopted type of resource discovery approach to address multi-attribute and range queries has been presented. Unlike the papers in this scope, this paper decouple resource discovery behavior model to data gathering, discovery and control behavior. Also it facilitates the mapping process between three behaviors by means of the formal verification approach based on Binary Decision Diagram (BDD). The formal approach extracts the expected properties of resource discovery approach from control behavior in the form of CTL and LTL temporal logic formulas, and verifies the properties in data gathering and discovery behaviors comprehensively. Moreover, analyzing and evaluating the logical problems such as soundness, completeness, and consistency of the considered resource discovery approach is provided. To implement the behavior models of resource discovery approach the ArgoUML tool and the NuSMV model checker are employed. The results show that the adopted resource discovery approach can discovers multi-attribute and range queries very fast and detects logical problems such as soundness, completeness, and consistency.

1. Introduction

In the past few years, with the extension of distribution systems such as the internet and web services as an information hub to facilitate data transfer and sharing (Rafat & Sher, 2013), also by increasing user requirements to compute with high precision and low cost, Grid computing systems emerged to be one of the utilizable solutions to reach these purposes. The term "Grid computing" was introduced in the mid1990s to aggregate the power of widely distributed resources for offering access to a vast collection of heterogeneous resources as a single, unified resource to solve large-scale computing and data intensive problems for advanced science and engineering (Foster & Kesselman, 1999; Jafari Balasangameshwara & Raju, 2012; Erdil, 2012; Navimipour & Khanli, 2008; Siva Sathya & Syam Babu, 2010). A fundamental service in the Grid computing is resource discovery which finds the appropriate resources for requested task to match the user's application requirements (Sarhadi, Yousefi, & Broumandnia, 2012). Also this service can be defined as searching and locating resource candidates which are suitable for executing jobs in a reasonable time in spite of dynamicity and large scale of the environment (Cokuslu, Hameurlain, & Erciyes, 2010). Allocating resources on Grid computing systems is a complex procedure involving multi-attribute and range queries, sharing and meeting the requirements of users and resource owners (Vanderster, Dimopoulos, Parra-Hernandez, & Sobie, 2009). Up to now, many resource discovery approaches in Grid computing are introduced and studied. These approaches generally are divided into five main categories (Jafari Navimipour, Rahmani, Habibizad Navin, & Hosseinzadeh, 2013): Centralized, Decentralized, Peer to Peer, Hierarchical, and Agent based approaches. Due to the advantageous of hierarchical approaches, in this paper an adopted type of tree-based mechanism based on (Khanli & Kargar, 2011) is introduced. Also to the best of our knowledge, all papers which studied the resource discovery in Grid systems, evaluate the proposed method only by simulation and experiments. Therefore, it is possible that some part of state space of problem is not analyzed. To solve this problem, model checking as an automatic technique for the verification of the software systems (Clarke, Grumberg, & Peled, 1999) is a suitable approach. Therefore, in this paper, model checking and formal

* Corresponding author. Tel.: +98 9144021694.
E-mail addresses: alireza.souri@iauasrb.ac.ir (A. Souri), jafari@iauasrb.ac.ir (N. Jafari Navimipour).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.11.042

verification of proposed resource discovery approach in Grid computing are presented.

Model checking is an automatic technique for the verification of the software systems (Clarke et al., 1999). It has several advantages over simulation, testing, and deductive reasoning. The model checking process includes the following steps: Modeling, Specification, and Verification. First, the model is generally built using an FSM-based formalism by a model-checking tool such as Spin (Holzmann, 1997) and SMV (McMillan, 1992). Then, a set of expected properties of the system define to satisfy these properties by using some temporal logic languages (Manna & Pnueli, 1992). After checking an expected property, there are two possible conditions: the property is satisfied by the model that means the model does not contain any misbehavior, and the property violated by the model means that one behavior in the model is not authorized by the specification. Then, the model checker tools generate a counterexample as output, which shows the behavior that violated the property (Dwyer, Avrunin, & Corbett, 1998).

The expression for mathematical demonstration of the correctness of a system is formal verification. Formal verification techniques have an important role in validation processes of systems. Formal verification of Resource Discovery approaches is essential as it can detect design flaws that lead to discovery failure. The most popular techniques are state-space searching and formal logics. State space searching involves exhaustive testing or scenario analysis.

In this paper, an adopted type of resource discovery approach to address multi-attribute queries has been presented in Grid computing. In particular, we separate resource discovery behaviors into three types: data gathering behavior, discovery behavior and control behavior based on behavioral modeling methods of Hansen, Virtanen, and Valmari (2003). The interactions between the three behaviors are modeled as characteristics process. The process role is to keep triple behaviors synchronized. By analyzing logical problems and checking behavior specifications, it is possible to verify the resource discovery approach. In particular, the contributions of this paper are:

- Proposing an adopted type of resource discovery approach based on tree structure to discover the multi-attribute queries.
- Presenting a resource discovery behavior model to decouple data gathering, discovery and control behavior of resource discovery approach. The isolation of resource discovery behaviors facilitates the procedure of maintenance, development, verification of resource discovery approach.
- Facilitating the mapping process between three behaviors by means of the formal verification approach based on Binary Decision Diagram (BDD) (Clarke et al., 1999). This formal approach extracts the expected properties of resource discovery approach from control behavior in the form of Computation Tree Logic (CTL) and Linear Temporal Logic (LTL) formulas, and verifies the properties in data gathering and discovery behaviors comprehensively.
- Analyzing and evaluating the logical problems such as soundness, completeness, and consistency of the considered resource discovery approach.
- Implementing the behavior models of resource discovery approach by using ArgoUML[1] tool and the NuSMV[2] model checker.

The rest of this paper is organized as follows. Section 2 reviews the related works and correlated studies in formal verification and resource discovery approaches. Section 3 presents an adopted resource discovery approach based on tree structure in Grid computing. Section 4 describes the data gathering, discovery and control behaviors as well as defining some essential concepts and notations to formalize these behaviors. Section 5 presents a symbolic model checking approach for the proposed behavioral models, and then the behavioral models convert to Kripke structure to check some behavioral problems such as completeness, soundness and consistency automatically. Furthermore, the logical properties of behavioral models are defined by using linear temporal logic and computation tree logic languages. These properties can be checked by the specification of control behavior which is mapped on data gathering and discovery behaviors. Section 6 focuses on the implementation of the proposed behavioral models by ArgoUML tool and verification of the Kripke structure model by NuSMV model checker. Finally, conclusions and future works are provided in Section 7.

2. Backgrounds and related works

Model checking is a popular technique for checking correctness properties of the supposed system, in which the design is represented as a finite state transition system, and the property is specified as a temporal logic formula (Ivančić, Yang, Ganai, Gupta, & Ashar, 2008). Up to now, many systems and applications are verified and checked with the help of these methods. In this section, first common resource discovery algorithms in Grid systems are reviewed; then some examples of the behavioral modeling and verifications are provided.

2.1. Resource discovery

The resource discovery approaches generally are divided into five main categories (Jafari Navimipour et al., 2013): Centralized (Benson, Wasson, & Humphrey, 2006; Fitzgerald et al., 1997; Kaur & Sengupta, 2007; Kovvur, Kadappa, Ramachandram, & Govardhan, 2010; Yu, Venugopal, & Buyya, 2003), Decentralized (Brocco, Malatras, & Hirsbrunner, 2010; Fouad, Syed Saadat, Abdolrezza, & Alexander, 2011; Iamnitchi & Foster, 2001; Jaehwan, Keleher, & Sussman, 2010; Kocak & Lacks, 2012; Li & Liu, 2007; Tangpongprasit, Katagiri, Kise, Honda, & Yuba, 2005; Zhu et al., 2004), Peer to Peer (Brunner, Caminero, Rana, Freitag, & Navarro, 2012; Caminero, Robles-Gómez, Ros, Hernández, & Tobarra, 2013; Erdil, 2012; Flocchini, Nayak, & Xie, 2005; Hawa, As-Sayid-Ahmad, & Khalaf, 2013; Iamnitchi, Foster, & Nurmi, 2002; Jagadish, Ooi, Tan, Vu, & Zhang, 2006; Jik-Soo et al., 2008; Marzolla, Mordacchini, & Orlando, 2007a; Marzolla, Mordacchini, & Orlando, 2007b; Mastroianni, Talia, & Verta, 2005; Mastroianni, Talia, & Verta, 2008; Maymounkov & Mazières, 2002; Meshkova, Riihijarvi, Petrova, & Mähönen, 2008; Sarhadi et al., 2012; Tan, Lü, & Lin, 2012; Trunfio et al., 2007), Agent based (Foster, Jennings, & Kesselman, 2004; Han & Berry, 2008; Jennings, 2001; Kakarontzas & Savvas, 2006; Kang & Sim, 2012; Lei, Xueyong, & Bei, 2010; Li & Li, 2011; Tan, 2009; Tan, Han, & Wu, 2010; Zhao, Yu, & Chai, 2007) and Hierarchical (Chang & Hu, 2010; Ebadi & Khanli, 2011; Elmroth & Tordsson, 2005; Khanli & Kargar 2011; Ramos & Melo, 2006; Yulan, Huanqing, & Xin, 2007) approaches.

In the centralized approaches a single or designated set of controllers discovers the resources which follow client/server architecture (Krauter, Buyya, & Maheswaran, 2002). In these approaches, the servers store information about the services which can be provided. When an entity requests a certain service, it sends a request to the server, and then finds appropriate resources and allocates them to the requester's entity. Since all query processes are done by a single or designated set of controllers, once a system exceeds several hundred nodes, the resource discovery approach

[1] http://argouml-stats.tigris.org/.
[2] http://nusmv.fbk.eu/.