Future Generation Computer Systems 37 (2014) 191–200

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Low complexity reconfiguration for real-time data-intensive service-oriented applications

Marisol García-Valls *, Patricia Uriol-Resuela, Felipe Ibáñez-Vázquez, Pablo Basanta-Val

Distributed Real-Time Systems Lab, Department of Telematics Engineering, Universidad Carlos III de Madrid, Av. de la universidad 30, 28911 Leganés, Madrid, Spain

HIGHLIGHTS

- Approach for the reduction of complexity in reconfiguration protocols.
- Suitable for reconfiguring dynamic distributed real-time SOA applications.
- Elaboration of the key phase for complexity reduction: *real-time prune algorithm* (RTPrune).
- Achievement of time bounded reconfiguration obtaining 90% reduction of graph complexity.
- Validation on a simulation tool built for the iLAND middleware.

ARTICLE INFO

Article history:
Received 15 May 2012
Received in revised form
24 August 2013
Accepted 13 October 2013
Available online 20 November 2013

Keywords:
Reconfiguration
Real-time
SOA
Distributed systems
Complexity reduction

ABSTRACT

Next generation distributed real-time systems will be complex high-performance environments containing applications with a flexible structure, integrating a large number of nodes of heterogeneous nature characterized by multiple and decoupled software units scattered all over the distributed environment; they are expected to offer data-intensive capabilities through merging the processing power of large numbers of nodes. These systems will have increased dynamic behavior by suffering frequent reconfigurations or state transitions resulting, among others, from the changing nature of the processed data. Handling the dynamics of these systems in real-time is a complex problem that requires to impose some bounds to the structure of the system to really achieve timely response not only during normal operation but also in the event of reconfigurations. In this paper, we present an approach to achieve real-time reconfiguration in distributed real-time service-based systems modeled as graphs. A reconfiguration requires to search for a new schedulable/valid solution or state from a complete system graph that contains all tentative solutions; each of these solutions will have undergo a schedulability analysis to determine if it is a valid solution; if the system graph is too complex, the overall time required for the schedulability check can be exponential with respect to the size of services and service implementations; this may lead to an unbounded reconfiguration time. In this paper, we present an approach to reduce the complexity of the system graphs so that a summarizing one that contains valid solutions is analyzed and not the complete system graph. We have implemented this mechanism inside the iLAND service reconfiguration and composition components to validate the proposed concepts and ideas; the reduction of the space of solutions with the presented approach is very high, which dramatically decreases the computation time of the reconfiguration process.

1. Introduction

Service Oriented Architectures (SOA) are today an active research field with significant progress in web-service (WS) based technologies. Distributed systems based on services increase the decoupling of components and increase the flexibility of the system as a whole in a way that new components and functionality pieces are easy to integrate compared to the monolithic deployments. In such a scenario, services are self contained functionality units that interoperate with other services by exchanging messages. They can be composed to construct enhanced functionality units called *applications*. In such a context, applications can suffer transformations by simply replacing one service by a different one or by a different implementation of the same service. In modern systems, this is a powerful characteristic that enhances the flexibility of systems at the cost of introducing dynamic behavior. These transformations are called *reconfigurations*, i.e., the process by which a system that is in a current state evolves to a

* Corresponding author. Tel.: +34 916248783.
E-mail addresses: mvalls@it.uc3m.es (M. García-Valls), puriol@pa.uc3m.es (P. Uriol-Resuela), fibanez@pa.uc3m.es (F. Ibáñez-Vázquez), pbasanta@it.uc3m.es (P. Basanta-Val).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.10.019

target state. Software-level reconfiguration has been traditionally approached from very different domains. The software engineering point of view has mainly presented a new component technology for easy (though not timely) replacement of functionality [1]; real-time theory has developed mode changing protocols and algorithms to ensure time-bounded transitions; more recently, service-oriented paradigms have focused on service composition although being silent about timing issues since their main target is still general purpose distributed systems. We target distributed real-time systems based on distributed services, and, in such a context, it is necessary to provide mechanisms for time-deterministic execution of the applications not only during normal operation but also during the reconfiguration process. For this, it is essential to eliminate or, at least, alleviate all sources of complexity in the reconfiguration process. The presented work is embedded in the context of the iLAND project [2] where we have developed an approach for time-bounded reconfiguration of distributed service-based applications. In iLAND, we define applications as graph based structures where nodes are the services and the links represent the message exchanged between them. Each service may have different *service implementations*, that are different versions of the same service providing the same functionality but with different output qualities and, as a consequence, different resource demands. A running application is a path of the graph of service implementations. As a consequence, reconfiguring an application consists of replacing one path by another. The reconfiguration process implies a selection of the new system configuration to make the transition to. This selection is performed according to both the schedulability or timeliness of the solution and the proximity to the desired application-level quality of service. We consider the end-to-end processing time of a distributed application as the Quality of Service value. Prior to the selection of the new state or solution, all tentative solutions have to be identified; after, they undergo a schedulability test. The new state will be selected from the schedulable solutions according to their proximity to the application-level QoS value. For the sake of efficiency and timeliness, the number of tentative solutions has to be as small as possible. One of the main problems is how to deal with complex graphs when real-time reconfiguration is needed. Searching through the complete space of solutions to obtain the new valid state to switch to can be unmanageable specially at run-time. Even with medium complexity graphs, it is desirable to reduce the space of solutions to search from. In this paper, we provide an approach to reduce the space of possible solutions that a reconfiguration process will consider to decide on the new state for making the transition. Firstly, we present the reconfiguration process and its logical break down into several time-bounded phases. After, the most complicated phase that is the reduction of the space of solutions of a graph is dealt with. Our approach is validated showing how we improve substantially the reconfiguration time. This paper is organized as follows. We present the related work in Section 2. Section 3 describes the main characteristics of the reconfiguration in distributed real-time systems. Section 4 presents our approach to reduce the graph complexity with a mechanism named *RTPrune*. Section 5 presents the validation of our approach. Section 6 concludes the paper and presents future work.

2. Related work

Service-based environments as described in iLAND [2] come from the traditional structuring of applications into functionality pieces with well-defined interfaces. Research on service-based paradigms has traditionally been applied to web-services [3,4], that is only a special part of service-oriented computing [5]. Web environments are typically not sensitive to real-time deadlines, but they are being used in projects that target industrial systems such as SOCRADES [6] where web services were applied to manufacturing and SIRENA and OASIS [7] where DPWS (Device Profile Web Services) middleware enables the interaction with the embedded nodes present at the factory floor. Also, other contexts for educational remote monitoring and control apply the service oriented paradigm [8,9]. In this context, timeliness is not handled from a real-time scheduling perspective but from an efficient low-level programming that yields acceptable performance results for embedded domains. Other work as [10] merged web services with CAMEX for industrial electronics devices applied to production environments. From a real-time perspective, some projects applied resource management in the context of distributed systems using SOA for interoperability among nodes. This is the case of [11], being limited about the schedulability of the complete system not considering the network effects. It does not aim at reconfiguration but rather it uses SOA for interoperability among nodes where a certain level of resource management and enforcement takes place. Also, project RTLlama [12] proposed a similar approach based on resource reservations and estimations of the response time of services; a centralized algorithm for service composition is used that is based on the calculation of the shortest path. Other approaches as [13] enhance the SOA model to provide a composition algorithm with the goal of supporting real-time applications. This approach does not consider the network effects as it uses a light real-time characterization which is not realistic in the domain.

Other real-time SOA efforts as [14] provide data transformation services that must be time-bounded. Their assumptions are the a-priori knowledge of the execution times of services, but the schedulability of the complete system is not considered, this contradicts the real-time guarantees. Therefore, reconfiguration is not a new concern. It has been studied in different domains but rather limited to centralized systems as [15–17] (in which a set of time-bounded steps were defined); or in the distributed systems domain, they were silent about timeliness as in [18,19]. Domains such as industrial systems have also received contributions for flexible architectures for safe execution and efficient data delivery (e.g. [20–22]). The majority of the analyzed related work proposes interesting solutions to web service composition for the general computing environments. However, they lack the adequate modeling of the real-time properties of services, the considerations of the schedulability of the system as a whole, the timely reconfiguration solutions, and the timely execution and communication guarantees for services. In this paper, we present an approach for overcoming the restriction of lack of timely reconfiguration by focusing on one aspect of the reconfiguration process that is essential to provide efficient reconfiguration in complex service-based systems; we present an approach for dramatically reducing the number of tentative solutions that must be checked for schedulability in the system; we provide a solution that is efficient in the transition and in the calculation of the target state based on the reduction of the complete space of possibilities. We identify the different reconfiguration phases, and we focus on the real-time pruning phase where the pruning of the space of solutions of the system takes place. Our previously presented reconfiguration contributions concentrate on in the iLAND middleware framework [2,23,9,24] providing low-level mechanisms for real-time mode changes as [16], providing resource management techniques based on priority management [17,25], time bounded server composition algorithms as [3], and a real-time schedulability framework based on budget scheduling as [26]; however, all of them leave aside the real-time pruning phase which is essential to achieve real-time reconfiguration, and that is presented in this paper.

3. Distributed real-time reconfiguration

3.1. Real-time service model and related components

Timely reconfiguration in distributed real-time systems is not a trivial task. For open distributed systems with lack of restrictions it is, in fact, not solvable with the available computing paradigms, i.e, a set of bounds and limitations to the structure of systems need to