The Journal of Systems and Software 82 (2009) 809–824

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Dynamic Web Service discovery architecture based on a novel peer based overlay network ☆

S. Sioutas [a,*], E. Sakkopoulos [b], Ch. Makris [b], B. Vassiliadis [c], A. Tsakalidis [b], P. Triantafillou [b]

[a] Department of Informatics, Ionian University, 7 Tsirigoti Square, 49100 Corfu, Greece
[b] Computer Engineering and Informatics Department, University of Patras, Greece
[c] Computer Science, Hellenic Open University, Patras, Greece

ARTICLE INFO

Article history:
Received 18 June 2008
Received in revised form 12 November 2008
Accepted 13 November 2008
Available online 3 December 2008

Keywords:
Web Services Discovery
Peer to Peer overlay networks
Databases

ABSTRACT

Service Oriented Computing and its most famous implementation technology Web Services (WS) are becoming an important enabler of networked business models. Discovery mechanisms are a critical factor to the overall utility of Web Services. So far, discovery mechanisms based on the UDDI standard rely on many centralized and area-specific directories, which poses information stress problems such as performance bottlenecks and fault tolerance. In this context, decentralized approaches based on Peer to Peer overlay networks have been proposed by many researchers as a solution. In this paper, we propose a new structured P2P overlay network infrastructure designed for Web Services Discovery. We present theoretical analysis backed up by experimental results, showing that the proposed solution outperforms popular decentralized infrastructures for web discovery, Chord (and some of its successors), BATON (and it's successor) and Skip-Graphs.

1. Introduction

The continuous expansion of the Internet and its relating technologies has created new marketing opportunities: traditional monolithic approaches in information system design are giving way to Service Oriented Computing (SOC). SOC supports the development of applications as if they were a connected network of functionalities (services) available, in a network-enabled environment, within and across different organizations (Singh and Huhns, 2005).

Web Services (WS), "a vision of loosely coupled interaction between components, programs, and applications" and a major implementation technology for SOC, is becoming a driver for business integration (Borenstein and Fox, 2004). Nevertheless, as more and more WS become available by many vendors, an old, search engine, problem is reappearing in a new form: searching (discovery) mechanisms of WS are not efficient both in response times and quality of results. Currently, the majority of Web Services are developed for internal enterprise use. Web Services will become more widely adopted in the years to come, allowing much broader intra- and inter-enterprise integration. It is anticipated that there will be an increasing requirement for automated service discovery, enabling further Web Services interaction with even less human effort. The 'information stress' problem in highly distributed environments is one of the most interesting and difficult for computer science, and has already attracted significant attention (Schmidt and Parashar, 2004; Yu et al.,2004a,b; Panagis et al., 2008; Diamadopoulou et al., 2008; Adamopoulou et al., 2007).

Web Service registries are helping to narrow down the negotiation and searching time needed for service discovery. Their basic concept lies in the matching mechanism of the contractual and technical profile of the query to that of the WS. In order for the searching procedure to be fast and more importantly automatic, information needs to be machine processable. Semantic Web technologies have already provided the first standards and tools towards machine to machine interaction; truly dynamic automatic discovery however is not yet in our grasp. WS technology is still categorised among most recent members of the Web Engineering area and it has already attracted both scientific community and vendor attention (Sycara et al., 2003). Promising steps have been made towards making WS a worldwide supported solution for application to application (A2A) interaction. Several recent technologies have been developed to support functionality in terms of service communication (e.g. SOAP), description (e.g. WSDL), and discovery (e.g. UDDI).

At present, Web Services are mainly advertised in catalogues which are based on the Universal Description, Discovery and Integration standard (UDDI). UDDI has become the predominant technological environment for WS Discovery. A large number of such centralized registry implementations for intra-enterprise communication, each focusing on registering services of interest

☆ Preliminary version of this paper was presented in Proceedings of IEEE International Conference on Information Technology: Coding and Computing, Next – Generation Web and Grid Systems (IEEE/ITCC 2005), pp. 193–198.

* Corresponding author.
E-mail addresses: sioutas@ionio.gr (S. Sioutas), sakkopul@ceid.upatras.gr (E. Sakkopoulos), makri@ceid.upatras.gr (Ch. Makris), bb@eap.gr (B. Vassiliadis), tsak@ceid.upatras.gr (A. Tsakalidis), peter@ceid.upatras.gr (P. Triantafillou).

0164-1212/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2008.11.845

to respective groups, are anticipated. As a result, apart from the number of different Web Services to select from, there is also the challenge of dealing with a large number of specialized, sector/business-oriented registries during service discovery and selection. Centralisation may lead to significant performance bottlenecks (Ouzzani and Bouguettaya, 2004).

A solution to the bottleneck problem that is gaining popularity among researchers is decentralized infrastructures based on P2P (Peer to Peer). The convergence of WS and P2P computing promises some exciting results (Papazoglou et al., 2003). Most existing decentralized discovery solutions in practice are based on the Chord (Stoica et al., 2003; Zhang et al., 2003) and BATON (Jagadish et al., 2005, 2006) infrastructures as well as the best theoretical solution refers to Skip-Graph (Goodrich et al., 2006).

This paper, based on the ideas initially expressed in Sakkopoulos (2005), presents a novel efficient and fault tolerant discovery-search infrastructure for P2P Web Service Discovery, called NIPPERS (Network of InterPolated PeERS). The proposed solution provides support for processing: (a) exact-match queries of the form "given a key, locate the node containing the key" and (b) range queries of the form "given a key range, locate the node/nodes containing the keys that belong to this range". NIPPERS is an extensive upgrade of the Interpolation Search Tree Data Structure (Mehlorn and Tsakalidis, 1993) for distributed P2P environment. Results from our theoretical analysis show that the communication cost of the query and update operations scale double-logarithmically with the number of NIPPERS nodes, outperforming three of the most popular decentralized structures, Chord (and some of its successors), BATON (and it's successor) and Skip-Graphs.

Furthermore, the system is robust with respect to failures, providing a solution in this way for (1) a fully decentralized registry environment, (2) a decentralized mapper of WSDL description files and/or XML query files for WS to keys stored in the overlay network peers, and (3) Quality of WS requirements.

The rest of this paper is structured as follows. Section 2 overviews related work in Web Services, P2P Computing and P2P Web Service Discovery. Section 3 includes details about implementing XML Web Service Discovery on P2P overlay networks and introduces the NIPPERS architecture. Section 4 analyses the proposed solution functionality. Section 5 discusses fault tolerance and WS QoS issues contributing a technique that increases the robustness of NIPPERS to failures. In Sections 6 and 7, implementation issues and the experimental evaluation of the proposed solution are presented respectively. Finally, Section 7 outlines items for future work and conclusions.

2. Theoretical background

2.1. Web Service management and standards

Web Services management may be considered as a threefold case involving description, discovery and interaction issues. Description is mainly about interface and functionality: what are the characteristics of a WS and how it works in order to solve a specific problem. Description is based mainly on WSDL (Web Services Description Language). Discovery is about finding the right WS for the right task and interaction describes how different WS cooperate with each other to perform that task. These three procedures are strictly interconnected. For example, the description of a WS highly affects the discovery mechanism: semantically rich descriptions increase the quality of query results and thus discovery efficiency (Sycara et al., 2003).

Security is also an increasing concern of the WS community. SOC implementations which use combinations of WS face more threats than usual systems because they inherit the security risks of the autonomous entities involved. This is natural since negotiation and coordination activities between WS take place over the network. WS security can be based on open standards for supporting integrity, confidentiality and user authentication.

Standards and protocols are of great importance for the wider acceptance of WS by the industry. Most protocols are generally purposed to transit into open standards but they have not yet reached a state of maturity (Singh and Huhns, 2005). WS standardization is still a complex process. Many vendors have announced their own versions, versions that serve somewhat different visions of what WS are and how they must be used. This has led to a variety of competing, conflicting, or overlapping protocols promoted by different standardization bodies and vendors. Organisations such as W3C, OASIS, WS-I, and Liberty Alliance and vendors such as IBM, Microsoft, Sun, Oracle and BEA are promoting different solutions. Table 1 summarizes a categorisation of the main approaches for WS standardisation.

WS discovery in particular, is based on registries that make available (advertise) information concerning functionalities, interfaces, creator, pricing, etc. The structure of these registries, initially proposed by OASIS and later gained wider acceptation, is based on UDDI. UDDI adopts a centralized, client–server model: WS are registered in a UDDI registry and clients are able to search it in order to find the appropriate solution to their problem. This method is already proved to be problematic in terms of efficiency and uniformity (Li et al., 2004; Ouzzani and Bouguettaya, 2004). The client–server model suffers from performance bottlenecks when too many users search the same registry at the same time. The classic approach to add more servers or use load balancing techniques is not an efficient (or cost effective) solution. Decentralized approaches to WS discovery have already been proposed by large software vendors: Microsoft's WS-Inspection is a mechanism that relies on XML description in order to search web sites for available WS and accompanying usage rules (Singh and Huhns, 2005).

Furthermore, the ever-increasing need for quality of WS (QoWS) provisioning complicates even more selection procedures. QoWS may be seen from different perspectives such as basic service provision (e.g. network latency) or functionality. The latter refers directly to WS discovery since a query may return more than one results that meet the functional requirements but provide different quality of service attributes. This problem resembles that of search engine persuasion: lots of results- low quality. This

Table 1
Standards used for Web Services management.

Operation	Standard
Process	
Publication and discovery	UDDI, USML, WS-Inspection
Orchestration	BPEL
Presentation	WSXL, WSRP
Description	
Definition	WSDL
User interface	WSUI
Business process design	XLANG
Interaction	
Messaging	SOAP
Security	
Representation of access control policies	XACML
Integrity and confidentiality	WS-Security (Web Services security language)
Legal	LegalXML
Negotiation	Web Services secure conversation language
Security policies	WS-SecurityPolicy
User authentication, attribute and authorisation assertion	SAML (security assertion markup language)