P2P Architectures for Semantic Service Composition

Angelo Furno, Eugenio Zimeo
University of Sannio
Department of Engineering
82100, Benevento - Italy

1. Introduction

Service discovery enables consumers to find, in a space of available services (i.e. service repository), the desired ones. Composition is useful when no simple service is able to satisfy user's needs by itself: (partially) ordered collections of services may instead be able to solve the requested goals. Each service (and service requests) can be represented as a transition from a set of states (pre-conditions) to another one (post-conditions). Describing Web services and user queries with semantics provides flexibility with automatic discovery and composition.

Centralized architectures cannot manage large and continuously growing spaces of services (the typical situation, nowadays) with reasonable resolution times. Hence, distributed architectures are required to implement efficient distributed service registries and effective discovery/composition capabilities. Standards like UDDI v.3.0[1] and ebXML[2] have partially addressed the problems of scalability and fault-tolerance by proposing selective replication to create hierarchical registry federations. However, hierarchical repositories may introduce benefits as regards resolution time only for resources identified by structured names. This is not the case of semantically described services, where semantic predicates are difficult to structure. In recent years, researchers have exploited P2P architectures to improve scalability, dynamicity and robustness of service discovery [1, 2] and composition [3, 4]. Existing proposals are typically based on three types of P2P overlays: unstructured, structured or hybrid.

Unstructured overlays, like in [1], assume registry nodes are organized into a random graph and use flooding or random walks to discover services available at overlay nodes. These approaches support arbitrarily complex queries and do not impose constraints on the node graph or on service placement. Thus, they improve flexibility, fault tolerance and semantic matching capabilities: any peer can publish its services in a local repository and semantic queries of the desired service may be effectively routed towards the right registry. However, they produce a high message overhead if proper strategies to reduce flooding are not used.

Structured overlays (like Chords [5]) impose constraints both on the service registry topology and on service publishing to enable efficient discovery. Each service has to be uniquely identified by a key. Nodes are organized into a structured graph and each key is mapped to a responsible node. These constraints provide efficient support for exact-match queries: they enable discovery of a service, given its key, in only $O(\log (N))$ hops [5]. Solutions like [4] adopt such kind of P2P overlays. However, semantics can be supported only in the form of exact matches, thus reducing the flexibility of semantics-based service discovery.

A special kind of (unstructured) P2P overlay is Semantic Overlay Network (SON) [6]. A SON is a virtual network, built on top of other overlays (which can be either structured or unstructured), designed to connect semantically related peers, i.e. peers that host semantically similar content.

To overcome the limitations associated with both structured and unstructured overlays, hybrid solutions that mix both the approaches have been recently considered.

The authors of [2] propose, in their recent work, a hybrid architecture that exploits Distributed Hash Tables (DHTs) with SONs for performing service discovery. The role of DHT is limited by the hash function used to map services to peers: it works only for exact matching and decides the peers where services are to be published. Consequently, the authors rely on SONs to outcome these limitations. However, they do not consider the service composition problem.

In our previous works [3], we proposed an efficient technique to improve discovery and composition in P2P-with-superpeers service networks, based on a probabilistic forwarding algorithm driven by network knowledge, such as network density, and traces of already discovered service compositions, represented by Composition Overlay Networks (CONs) handled by superpeers.

Now, we are working on defining efficient and effective P2P-based solutions for semantic service composition, based on different kinds of overlay networks in order to analytically and experimentally compare the advantages and disadvantages (and the conditions when they apply) coming from the use of DHTs, SONs or unstructured (CONs-based) P2P architectures.

2. Semantic Service Composition with DHTs, SONs and CONs

Discovering a service that is not present as an atomic service description in a distributed registry demands for several related searches over the network to find all the composing services in the right order. This problem changes, if compared to simple discovery, the analysis of the benefits of structured and unstructured P2P overlays.

In Table 1, we propose an analytical comparison among possible DHT-based (Chord topology), SONs and CONs solutions for semantic service discovery and composition, showing their advantages and disadvantages. The results are based on the considerations reported on below and extend the ones in [2, 3].

The first problem we are investigating regards how to implement automatic semantic service composition over P2P networks, i.e. to find *k*-length sequences of services able to satisfy the user-specified goals.

[1] http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf
[2] http://www.ebxml.org

ISBN: 978-3-902457-39-4

Proceedings of the Work in Progress Session of the 22nd Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP 2014)

We start from unstructured overlays, to exploit a superpeers-based architecture (see [3, 7]). In case of SONs, the same backward collaborative search algorithm described in [3] can be used to perform multiple semantic searches over the available SONs to find composite solutions. In SONs, discovery time depends on the overlay topology, on the semantic matching similarity function and on the forwarding algorithm (typically flooding), while, with CONs, discovery cost is optimized using our informed selective forwarding strategy (see [3]).

Semantic service composition in a DHT-based system may be effectively performed if the following problems would be solved: (1) definition of an effective hash function able to identify the services with the desired semantics; (2) decentralized coordination for handling concurrent searches of component services (the query issued by a node should be propagated to the nodes that have the partial solutions); (3) low churn-handling complexity in a domain context where services leave and join the system frequently.

Regarding coordination, composition in DHTs may be implemented in two different manners: (1) *orchestrated*, in which composition processing is performed only by the submitter node, considering that the query is not propagated to other peers; (2) *cooperative*, in which the query is propagated to each looked-up peer, which in turn is able to continue the search for the other component services.

In the orchestrated approach, the overall cost to find M k-size service compositions (i.e. composition cost) is $k \cdot M \cdot DiscoveryCost_{DHT}$.

In the cooperative approach, the overall cost to find M k-size service compositions depends on the distribution of the component services in the network.

In fact, since in a DHT one single peer (P_i) hosts the set S of services that match the query at the i^{th} composition step (i.e. with the same post-condition in our case), that peer is responsible for solving the remaining partial queries. For each service in S, a partial query having the service pre-condition as its post-condition can be defined.

If the partial queries of the $(i+1)^{th}$ composition step have all the same post-condition (worst case), the hash function returns one single peer again (P_{i+1}) which is responsible in turn of solving the remaining part of the queries. At the end, if M different compositions exist in the system, the overall composition cost will be $k \cdot M \cdot DiscoveryCost_{DHT}$.

On the other hand, if the partial queries of the $(i+1)^{th}$ composition step have different post-conditions (best case), the hash function returns M different peers that are able to solve the remaining part of the queries in parallel. At the end, if M different compositions exist in the system, the overall composition cost will be $k \cdot DiscoveryCost_{DHT}$.

Therefore, in general, the overall composition cost is $k \cdot C \cdot DiscoveryCost_{DHT}$, with $1 \leq C \leq M$.

In CONs, composition cost is equal to the CON discovery cost if the compositions already exist in the overlay. In fact, since they are available in superpeers' repository, it is only necessary to reach one of the peers in the overlays to get access to the superpeers and thus to the available solutions. If the compositions do not already exist in the overlay (worst case), the overall composition cost will be $k \cdot DiscoveryCost_{CON}$, because M searches for k-size service chains will be performed concurrently over the underlying unstructured topology (each one costing $k \cdot DiscoveryCost_{CON}$).

Arch. / Property	DHTs	SONs	CONs
Content placing	Fixed, based on hashing	Each peer is responsible for its content	Each peer is responsible for its content
Content retrieval	Exact match	Semantic (based on similarity)	Semantic (based on similarity)
Network structure	Fixed	Variable	Variable
Discovery cost	$O(log(N))$	Variable	Variable, but optimized
Topology management	Stabilization protocol	Semantic links need to be rearranged	Composition semantic links need to be rearranged
Semantics	None	Exploit a semantic artifact (e.g. ontology)	Exploit a semantic artifact (e.g. ontology) for atomic and composite service matching
Composition cost (for M *k-service* compo-sitions)	$k \cdot M \cdot DiscoveryCost_{DHT}$ (orchestrated) or $k \cdot C \cdot DiscoveryCost_{DHT}$ with $1 \leq C \leq M$ (cooperative)	$k \cdot DiscoveryCost_{SON}$	$k \cdot DiscoveryCost_{CON}$ (worst case) or $DiscoveryCost_{CON}$

Table 1: DHT(Chord), SON and CON-based semantic service composition

We have already validated our approach in [3] and compared it with other gossip-based forwarding strategies in [7], by using the PeerSim [8] simulator to observe resolution time, recall and message overhead on small and large-size P2P networks. We are currently working to the implementations of the proposed DHT-based and SON-based solutions in the simulator, in order to verify our analysis by means of several experiments over different network topologies (for both the structured and unstructured approaches), network sizes and peer churn rates.

3. References

[1] M. Bisignano, G. Di Modica, and O. Tomarchio, "JaxSON: A semantic P2P overlay network for web service discovery," in *Proc. of IEEE World Conference on Services-I*, 2009, pp. 438–445.

[2] G. Pirrò, D. Talia, and P. Trunfio, "A DHT-based semantic overlay network for service discovery," in *Future Generation Computer Systems*, vol. 28, no. 4, pp. 689–707, 2012.

[3] A. Furno and E. Zimeo, "Efficient cooperative discovery of service compositions in unstructured P2P networks," in *Proc. of the 21th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pp. 58–67, 2013.

[4] Z. Zhengdong, H. Yahong, L. Ronggui, W. Weiguo, and L. Zengzhi, "A P2P-based semantic web services composition architecture," in *Proc. of IEEE International Conference on e-Business Engineering*, 2009, pp. 403–408.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM Computer Communication Review*, vol. 31, 2001, pp. 149–160.

[6] A. Crespo and H. Garcia-Molina, "Semantic overlay networks for P2P systems," in *Agents and Peer-to-Peer Computing*. Springer, 2005, pp. 1–13.

[7] A. Furno and E. Zimeo, "Gossip strategies for service composition," *22th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, 2014 [to appear].

[8] A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator," in *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*, September 2009, pp. 99–100.

ISBN: 978-3-902457-39-4