Service Searching based on P2P Aggregation

Adriano Fiorese[1,2], Paulo Simões[1], and Fernando Boavida[1]
[1]CISUC – Dep. Eng. Informática
Pólo II da Universidade de Coimbra
3030-290 Coimbra, Portugal
{fiorese, psimoes, boavida}@dei.uc.pt

[2]Computer Science Department – DCC
University of the State of Santa Catarina – UDESC
89223-100 Joinville, Brazil
fiorese@joinville.udesc.br

Abstract— **This paper presents a Peer-to-Peer approach to the problem of searching service components in multi-domain environments. This approach overcomes a number of limitations associated with classical approaches, when applied in multiple administrative domain scenarios, improving scalability, resilience and performance. An evaluation of the proposed framework, focusing on performance and scalability, is also provided.**

Keywords – Multi-domain services, P2P Aggregation Services

I. INTRODUCTION

The access to interconnected computing resources is an ever-growing necessity. Computational services are currently indispensable and new requisites and necessities arise every day. These needs are leading to the development of new and more complex services, which often involve the active cooperative collaboration of multiple providers (connectivity, content, data processing, storage, identity management, etc.) belonging to distinct administrative domains.

Several management activities are involved in the offer of a service that covers multiple domains. One of these activities is the search for services that have the potential to be used as components in the dynamic composition of more complex services offered to end-users. These components might include content (e.g. finding vendors offering a specific movie end-user wants to see; finding dubbing or subtitling services for that same movie), connectivity (e.g. interconnection links with QoS and security requirements between the consumer and the origin of the contents) and complementary services (electronic billing and/or payment systems; multi-session controllers for video-conferencing sessions, etc.).

In an open and dynamic environment such as the Internet, it is possible to foresee a scenario where several providers compete among themselves to offer these components to the entity that, in behalf of the final user, subcontracts them and coordinates the composition and lifetime management of the assembled service. One of the first challenges faced by dynamic, on-demand construction of such composed services is to seek and locate candidate components in non-structured large-scale environments. Such challenge still lacks satisfactory solutions.

In this paper we propose the so-called Aggregation Service (AgS), a framework for service search and location based on Peer-to-peer (P2P) technologies and well suited for the already discussed multi-domain scenarios. AgS provides support for (i) publishing service offerings and associated management mechanisms, by interested service providers; and (ii) search operations based on flexible criteria. The P2P approach followed by AgS is well tailored to meet the multiple-domain, large scale scenarios mentioned above because it does not depend on dedicated supporting infrastructure (such as servers and storage), it has high scalability and it is flexible enough to handle the registries of services in a distributed manner.

This paper is organized as follows: Section II discusses related work and Section III provides a description of the AgS framework. Section IV is dedicated to the evaluation of the proposed framework, by means of simulation studies focused on performance and scalability. Finally, Section V concludes the paper.

II. RELATED WORK

The problem of network and services management in multi-domain environments is not new. Soukouti and Holberg [1], for instance, already suggest in 1997 to integrate management protocols with the distributed objects searching functionalities provided by the CORBA technology. However, the CORBA naming service imposes a significant overhead on its configuration in order to adequately support multi-domain environments, which means this solution is not very dynamic or very scalable.

Nowadays, web services seem to be the most popular solution for offering service interfaces. Several authors have addressed the use of P2P for discovery of web services. Schmidt and Parashar [2], for instance, propose a structured P2P overlay, based on CHORD [3], to improve the search of web services by using multiple keywords. This approach introduces a mapping of the web services definition keyword space in the P2P overlay search key space by means of SFC[4]. An attempt to include semantics in the description of web services, allowing them to be classified at the registries (peers) by means of an ontology-based approach is made in [5]. This work uses a structured P2P overlay among the registries to facilitate the search. The authors of [6] propose a semantically

This work was partially funded by FCT, under scholarship contract SFRH/BD/45683/2008.

enhanced service discovery-based on an unstructured P2P overlay. All these proposals claim that the P2P approach has some advantages for the service discovery process, when compared with centralized approaches such as UDDI [7]. These advantages include scalability; enhanced failure tolerance (by eliminating the single point of failure of the centralized approaches); and efficiency (by reducing the overhead of updates and replication operations that can be found in centralized approaches).

Other authors investigating P2P mechanisms for network and services management [8][9][10] propose cooperation among network managers by mutual P2P-based exchange of messages, and also by allowing the management tasks to be carried out by groups of peers, which can also play an important role in autonomous management contexts, including self-healing.

The mechanisms outlined in this paper fit in this trend of applying P2P techniques as a way to overcome some of the limitations imposed by current management approaches when applied to large-scale, highly dynamic multiple-domain environments. More specifically, we propose to create a two-tiered P2P overlay for searching service components. This two-tier approach allows faster searches with lower computational costs, through the aggregation of the service offerings in the higher tier. In the next section, there will be an examination of the main characteristics of the proposed AgS framework, which will be later assessed in Section IV.

III. THE AGGREGATION SERVICE

The Aggregation Service (AgS) we propose is a unstructured P2P overlay consisting of peers that represent potential providers offering their services (which may potentially make part of service composition in a cooperative /competitive way). Each service offer may be represented by more than one peer, which allows for some redundancy. The peers that form the AgS P2P overlay are called *aggregation peers*. The purpose of the aggregation peers is to aggregate the offerings of services and service components in an overlay tier.

The AgS also includes another class of peers: the *management peers*. These also belong to the service provider offering involved service components. Unlike the aggregation peers, management peers participate in the AgS by making the administration and management interfaces of the services indirectly available to external entities (such as service aggregators), likely located in other domains. Such management interfaces are used in the assembly and lifetime management of the composed services. Management peers make these interfaces available by publishing a service offer at several aggregation peers in the AgS P2P overlay. These aggregation peers may be located in the same domain as the management peers or, in some cases, in different domains.

Searching for a service, using the AgS framework, therefore results in a set of references for management peers that offer an interface to services matching the search criteria. This preserves the internal details of the service, since the external entity is only granted with a mediated access (by means of the management peer), which may hide sensitive information and filter undesired operations.

The AgS can be used as a non-structured solution since it can be deployed within the current infrastructure of the service providers without increasing costs. In addition, the two-tiered AgS architecture allows the publishing and searching functionalities (provided by the AgS) to be separated from the management functions to be carried out by the management peers. Thus, the manager of the service provider domain can restrict the sensitive configuration of its services (e.g. the existing management services, topologies, etc.) by only making available a previously selected set of interfaces for services and service components to a third party.

The AgS architecture, including the links between the aggregation service tier and the service management tier, is presented in Figure 1.

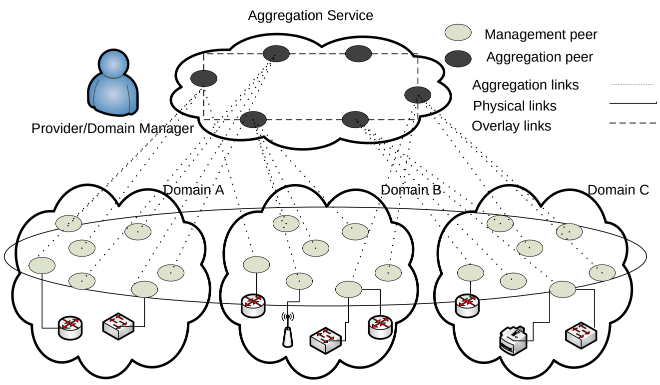

Fig. 1 The Aggregation Service Architecture

The design of the AgS depicted in Figure 1 consists of a P2P overlay using a ring topology, although other P2P topologies can be easily applied. Figure 1 also shows the management peers belonging to different administrative domains, which announce their services and service components to the aggregation peers.

A. Usage Scenario

The way the AgS is used by the provider of a composed service is very simple. This provider uses the AgS to search the management peers where the required service components are available. Taking Figure 1 as a reference-point, let's consider that a service provider from Domain A subcontracts services located in Domains B and C (e.g. a connectivity service with QoS guarantees and access to multimedia content). In this scenario the AgS ensures that the management peer reference (the search process) for this service can be recovered. Further access to the service interfaces (contracting, monitoring and life cycle management, etc.) is provided by the management peer that represents that service component. Interaction between the service contractor and the management peer is conducted outside of the AgS. A specific usage scenario can be found in the Introduction section of this paper. There, the on-demand stream-like movie delivery with its associated service components is overviewed.

B. Operations

The activities of the AgS are based on a number of operations. Table I presents its key operations and the corresponding messages exchanged among peers.

IV. VALIDATION FRAMEWORK

In order to assess the AgS behavior, we conducted a simulation study to measure the average path length necessary to locate the management peer that makes a particular service available. The average path length acts as a metric to evaluate the effectiveness of the AgS. The simulation involved 1,000 aggregation peers, which published the services that were made available by 10,000 management peers spread over 10 different domains. For the sake of simplicity, a particular management peer can only offer, at most, seven services from randomly chosen component services (using a uniform distribution) from the service set S={S1,S2,S3,S4,S5,S6,S7}. Each management peer can only publish its service subset on, at most, 10 distinct, randomly chosen, aggregation peers (also following a uniform distribution). Nevertheless, it is possible that more than one management peer can offer the same services subset at the same domain and publish it on the same aggregation peer. In the interests of simplification, the search concludes with the first match (although AgS also supports the return of all matches).

The simulations run for 50 hours. Each simulation used the same scenario and a different number of search operations were simulated, which ranged from 100 to 1,000 operations. Every operation (e.g. joining, leaving, publishing, and searching) is discretised in time. Figure 2 displays a fragment of the operations file that feeds the simulation.

The simulations run on a "well-behaved" scenario that is also described as consistent. This means the execution of the operations follow a temporal sequence. As depicted in Figure 2, the Publish operation only executes when every aggregation peer on the management peer's publication list has already carried out the Join operation. The same happens in the case of the Leave operations. The Query operations however do not follow this pattern. Thus, for example, the search operation conducted by A628 has as result M86, if any of the aggregation peers (A593, A67, A222, A797, A134, A433) had not yet left the AgS P2P overlay. In this case, there would only be a QueryReply message to A628 if some other management peer belonging to Domain 6 had made its publication of the S7 service to another aggregation peer that had not yet left the overlay.

```
A433 497m join
A134 721m join
A593 777m join
A67 1402m join
A797 2734m join
A222 2813m join
M86 2921m ManagementNodesApplication:publish
A134 1342m leave
A433 1361m leave
A67 2422m leave
A797 2770m leave
A593 2818m leave
A222 2861m leave
A628 2937m searchService DHCP@Domain6
```

Fig. 2 Fragment of the simulation operations file

A. *The Simulator*

The PeerFactSim.KOM[11] simulator was used in the simulations. This simulator is based on discrete events, which explains why the operations file is used. It is particularly designed to simulate large scale P2P networks.

TABLE I. TABLE OF OPERATIONS AND MESSAGES

Operation	Objective	Executor	Message sent
Join	Form the Aggregation Service.	*aggregation peer*	JoinMessage sent by the requesting peer to its successor and predecessor in the overlay.
Leave	Leave the Aggregation Service (in a normal way).	*aggregation peer*	LeaveMessage sent by the requesting peer to its successor and predecessor in the overlay.
Query	Look for a service/service component peer that provides a particular service in a specific domain.	*aggregation peer*	QueryMessage sent by the requesting peer to its successor in the overlay ring in a clockwise manner. The message is forwarded clockwise until it arrives at its goal or until the message reaches the requesting peer. When the desired information is found, a QueryReply message containing it, is then created. This latter message is directly transmitted to the requesting peer of the Query´s operation.
Publish	Make the services to be searched available.	*management peer*	PublishMessage sent by the management peer to its aggregation peer(s), which makes the service(s) public. A PublishMessage is sent to each aggregation peer.

B. *Results*

Two environments were simulated using the aforementioned scenario: (1) an environment based on the replication of the search results at the search requesting aggregation peer, and (2) an environment that was not based on replication of the search results.

Figure 3 depicts the relation between the total number of simulated operations and the number of query operations. It is possible to observe a query hit ratio of almost 100%, i.e., almost every searched service or service component is found and made available by at least one management peer.

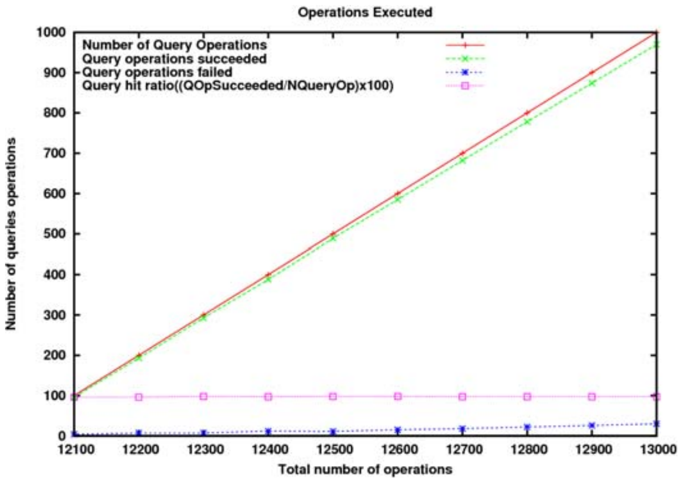

Fig. 3 Operations vs. Query Operations

As already mentioned (see Table I), the Query operation generates an initial message that is successively forwarded until it obtains the desired information or finds the requesting aggregation peer. Each transmission/forwarding represents a hop on the message path. Thus, the average path length will be the ratio between the number of hops and the number of search messages or search operations that are successfully or unsuccessfully accomplished, depending on the type of analysis that is required. Equations 1 and 2 show, respectively, how path length and average path length are calculated.

$$PL = \sum_{hops=0}^{n_{infoFound}} hops \qquad (1) \qquad\qquad APL = \frac{\sum_{hops=0}^{n_{infoFound}} hops}{QueryReplyMessages} \qquad (2)$$

Figure 4 shows the number of search messages sent; the number of successful answers to the search messages (query replies); and the number of successful and unsuccessful search operations. In addition, it depicts the number of hops that were undertaken in the successful and unsuccessful search messages. On the basis of Figure 4, it is evident that some of the search operations (Query) have not sent any messages, and this can be corroborated by the difference between the number of Query operations successfully completed and the number of QueryReply messages. This difference can be explained by the fact that some of the aggregation peers that began the search are the same that keep the information that is being searched. Moreover, it can be observed that the number of hops in unsuccessful search messages is lower than the number of hops of successful search messages. This is not unexpected since the number of unsuccessful Query operations is also low.

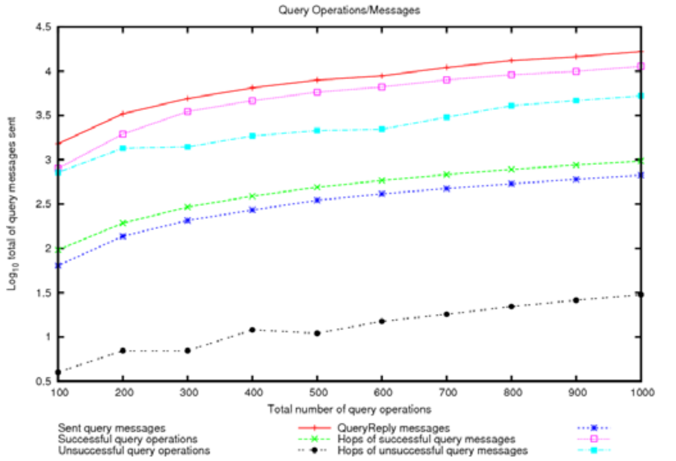

Fig. 4 Operations, Messages and Message Hops

Although the number of hops in unsuccessful query messages is lower than in successful query messages, the average path length is higher. This can be seen in Figure 5, that depicts the average path length for the environment with replication of the search results. This graph allows a comparison to be made between the average path length in unsuccessful searches and the average path length in successful searches.

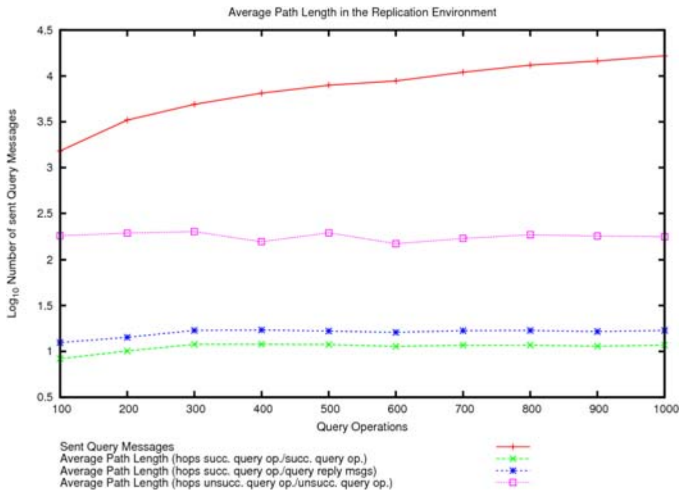

Fig. 5 Average Path Length in the Replication Environment

Figure 6 compares the results obtained from the preliminary analysis of the average path length for both simulated environments (with and without the replication of the query results). It can be seen that in the cases where the Query operation was unsuccessful the replication does not alter the average path length, because the queried service was not found. A comparison between the curves representing the average path length, with and without the replication of the results, shows the benefits of the replication. Moreover, this comparison also highlights the fact that some Query operations were successfully executed, although without the corresponding QueryReply message.

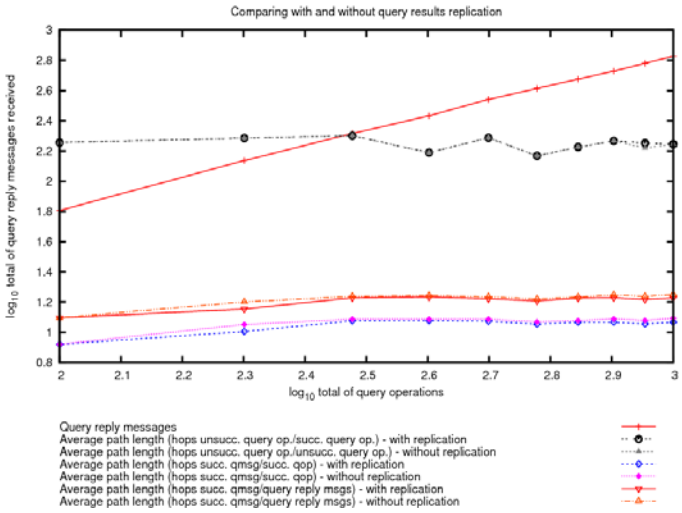

Fig. 6 Comparing Average Path Length in both the Environments: With and Without the Replication of the Search Results

These preliminary results suggest that the average path length depends on the number of active aggregation peers at the AgS P2P overlay. Although the number of simulated aggregation peers was 1,000, the AgS P2P overlay consisted, on average, 177 active aggregation peers with standard deviation of 16. This is the number of hops undertaken by unsuccessful query messages that traveled through the entire overlay and did not find the desired information. This low number of active aggregation peers can explain the low average path length that was shown up by the results. Considering that, the AgS seems to be scalable. The average path length is low and a high rate of successful queries is obtained, even when the number of Query operations varies.

Another important finding was that the replication of the results improves the AgS performance. The replication reduces the average path length, which means saving bandwidth (i.e. fewer hops are needed to reach the destination…), and savings in processing (fewer accesses to local caches, and fewer messages being forwarded…). This improves the performance with regard to the queries for the same service components.

However, a further study needs to be conducted with a greater focus on reducing the time constraints on the operations. A comparison between both the simulated scenario outlined above and a more relaxed one, could provide us with clues about the behavior of the Query operations in a high-churn environment.

As a part of a future work, we plan to address the maintenance of the AgS data consistency. This improvement is important to prevent the occurrence of query results that do not actually represent active services. A possible approach is to send a message (e.g., ACK message) to the management peer found by the query operation to verify the component service availability. This approach may prevent the overhead of the management peers' pooling monitoring messages.

V. Conclusion

This paper presented AgS, a framework based on a P2P overlay designed for the search of services in non-structured, large-scale environments involving multiple administrative domains.

This framework is based on the publication, in a P2P overlay, of service offers. Searching for services will result in the location of corresponding service interfaces (provided by management peers), that can be used by external entities to assemble composed services for end users. Management peers mediate the access to the offered services. This allows the enforcement of access policies and security measures, as well as the encapsulation of sensitive internal technical information.

In this paper we also assessed AgS. Two environments were simulated: first, an environment using the replication of the search results at the aggregation peer where the query was initiated; second, an environment where no replication of the search results was used. The assessment drew on the average path length at the search of service components as a metric to evaluate the effectiveness of AgS in terms of scalability and performance.

References

[1] Soukouti e Hollberg, "Joint inter domain management: CORBA, CMIP and SNMP," Proceedings of the fifth IFIP/IEEE international symposium on Integrated network management V : integrated management in a virtual world, London, UK: Chapman & Hall, Ltd., 1997, pp. 153–164.

[2] C. Schmidt e M. Parashar, "A Peer-to-Peer Approach to Web Service Discovery," World Wide Web, vol. 7, Jun. 2004, pp. 211-229.

[3] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, e H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, California, United States: ACM, 2001, pp. 149-160.

[4] H Sagan, Space-filling Curves, New York, NY, USA: 1994.

[5] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, e J. Miller, "METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services," Information Technology and Management, vol. 6, Jan. 2005, pp. 17-39.

[6] F. Banaei-Kashani, C. Chen, e C. Shahabi, "WSPDS: Web Services Peer-to-peer Discovery Service," IN PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON INTERNET COMPUTING, 2004, pp. 733--743.

[7] L. Clement, A. Hately, C. von Riegen, e T. Rogers, "UDDI," UDDI Version 3.0.2.

[8] L.Z. Granville, D.M. da Rosa, A. Panisson, C. Melchiors, M.J.B. Almeida, e L.M.R. Tarouco, "Managing computer networks using peer-to-peer technologies," Communications Magazine, IEEE, vol. 43, 2005, pp. 62-68.

[9] C.C. Marquezan, C.R.P.D. Santos, J.C. Nobre, M.J.B. Almeida, L.M.R. Tarouco, e L.Z. Granville, "Self-Managed Services Over a P2P-Based Network Management Overlay," 2nd Latin American Autonomic Computing Symposium (LAACS 2007), Petropolis - Rio de Janeiro - Brazil: 2007, p. 7.

[10] dos Santos, Santa, Marquezan, Cechin, Salvador, Granville, Almeida, e Tarouco, "On the design and performance evaluation of notification support for P2P-based network management," SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, New York, NY, USA: ACM, 2008, pp. 2057–2062.

[11] A. Kovacevic, S. Kaune, N. Liebau, R. Steinmetz, e P. Mukherjee, "Benchmarking Platform for Peer-to-Peer Systems," it - Information Technology, vol. 49, 2007, pp. 312-319.