Performance Evaluation of Service Searching using Aggregation in Peer-to-Peer Service Overlay Networks

Adriano Fiorese*†
*UDESC, University of the State of Santa Catarina
CCT, Department of Computer Science - DCC
89223-100, Joinville-SC, Brazil
e-mail:fiorese@joinville.udesc.br

Paulo Simões†, and Fernando Boavida†
†CISUC, University of Coimbra
Pólo II, Department of Informatics Engineering - DEI
3030-290 Coimbra, Portugal
e-mail: {fiorese,psimoes,boavida}@dei.uc.pt

***Abstract*—This paper presents a performance evaluation in the context of the Aggregation Service (AgS). The AgS is a P2P overlay-tier whose purpose is to aggregate the services and service components maintained by service providers in a P2P Service Overlay Network (SON). The performance evaluation takes into account two metrics: 1) the average path length, and 2) the response time. Both of them are used in the comparison of two environments: 1) AgS and 2) a P2P SON without AgS. Additionally, the searching performance in the environment with AgS is compared with a P2P SON that uses Gnutella as the searching mechanism. The simulation results clearly show an improvement in the performance of the search operations when the AgS is used to the detriment of the searches performed without it. The results also show AgS is better suited for use in small overlays.**

***Index Terms*—services management, P2P, service aggregation**

I. Introduction

Currently, the Internet is becoming the drive force for new businesses. Some of these new businesses rely on the concept of services as the elements that make the business players moving forward since services are the products sold to final users. Thus, these services and service components may include content (e.g. finding vendors offering a specific movie; finding dubbing or subtitling services for that same movie), connectivity (e.g. interconnection links which satisfy 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.). These services might be dispersed among several different providers on the big computing cloud the Internet currently represents. Such services and service components need to be searched, grouped, composed, provisioned, etc., in order to offer the final users a consolidated new product (service).

A possible approach to allow service providers to execute the necessary operations to make a service ready to be offered to final users is the creation of a Service Overlay Network [1]. The purpose of such overlay network is to span all the service providers that want to offer their services and service components. This approach has the advantage of sharing the offering costs among all the participants. Also, it enhances the service providers ability to make their services or service components available.

In a foreseen scenario where the number of services; component services and also service providers get continuously growing up, such as in the Future Internet, new mechanisms beyond the SON are needed in order to optimize the search of these service and service components.

This paper deals with the process of evaluating the performance and scalability of the Aggregation Service (AgS) framework proposed in [2]. It extends and improves the preliminary assessment performed in [3] by adding the evaluation of the response time and also comparing the AgS results with a Gnutella P2P SON.

Having in mind the stated goal and approach, this paper is organized as follows. Section II discusses related work. Section III shortly describes AgS. Subsequently, Section IV describes the simulated scenarios; presents and discusses the simulation results. Finally, Section V summarizes the paper and discusses further work.

II. Related Work

A. Network and Services Management

Currently, web services are the most developed approach to network and services management [4]. Also, they are the most popular solution for offering service interfaces and service composition, on which the Service Oriented Architecture (SOA) is built. Therefore, the searching of web services is a recurrent challenge. Work in this area comprises how to select and represent information about web services, as well as ways to overcome the limitations of the single centralized Universal Description, Discovery and Integration (UDDI) repository. Among others, proposals in this area includes searching web services by their operations based on the similarity of the desired operation [5].

B. Peer-to-Peer and Cross-Domain

The P2P overlay networks are significantly used as supporting-tier application. Beyond the traditional file sharing applications, resource discovery is commonly executed by these overlays. Michel et. al. [6] proposed the exploitation of keywords and attribute-values co-occurrences for the improvement of keyword-based searching in P2P. An intelligent resource discovery mechanism based on weaving attributes in indices using locality sensitive hashing and performing search based on the geographic location of the indices in a structured P2P overlay is presented in [7].

The searching by combining Grid and P2P was also proposed [8], [9]. Some ideas on this area concern the use of routing indices and mechanisms to easily spread them to the Grid; the utilization of bio-inspired algorithms in order to achieve overlay self-organization and selective flooding for the search exploiting particular conditions on local caches. The cross-domain service discovery is well studied in [10].

C. Service Overlay Networks

A Service Overlay Network (SON), is a virtualized network composed of interconnected nodes, whose generic purpose is to provide the required Quality of Service (QoS) to applications that execute on those nodes [1].

A P2P overlay network can also provide QoS services. We claim this can be accomplished when the participants are in a consortium of service providers that establish well-defined SLAs to regulate the contribution of each participant to the network. In this sense, these particular P2P overlay networks can be considered SON.

Lavinal et. al. [11] also uses P2P as support for the SON architecture. In that piece of work the authors also address the discovery of services, although they consider QoS aspects in their approach whilst we take into account performance aspects.

III. Searching with Aggregation Service

The Aggregation Service (AgS) is an unstructured P2P overlay-tier that executes on top of a P2P SON composed by service providers. AgS is composed of peers that belong to these service providers committed to offer their services in large scale. Therefore, the AgS optimizes the service and service components searching in a multi-domain environment composed by multiple service providers organized in a common P2P SON.

The purpose of the AgS is to aggregate service and service components. This is accomplished by concentrating the service offerings in its peers (nodes), in order to facilitate and optimize the searching. Each peer involved in the AgS can take care of several service offerings. A single service offering can be spread over multiple AgS peers in order to allow some redundancy and to overcome churn. Service providers can make available the peers that compose the AgS or rely on third party P2P overlay network providers.

The peers that form the AgS P2P overlay are called *aggregation peers*. They are responsible for keeping the service offering published on them, and to execute the searching process. The peers that compose the P2P SON are called *SON peers*. These peers form the overlay network where the true services and service components actually execute. SON peers make service interfaces available by publishing a service offer at several aggregation peers in the P2P AgS overlay. The service offering also keeps information about which SON peer is publishing it. The aggregation peers may be located at the same domain as the SON peers or in other domains.

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

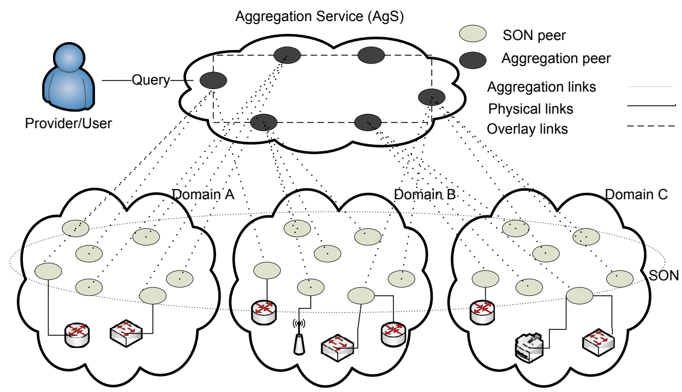

Fig. 1. Aggregation Service Architecture

IV. Evaluation Framework

Our evaluation framework relies on the response time (RT) for the query message to locate the SON peer reference that makes a particular service available. A simulation study was conducted in order to get values for this metric. The simulations involved a sample of thirty particular sets of aggregation peers. Each particular set kept the services that were made available and published by 10,000 SON peers spread over 10 different domains.

For the sake of simplicity, a particular SON peer can only publish, at most, seven services or service components randomly chosen (using a uniform distribution) from the service set S={S1,S2,S3,S4,S5,S6,S7}. Each administrative domain has its own set S. Also for the sake of simplicity, each service or service component profile is treated as a single service descriptor consisting of the name of the service or service component followed by and its particular domain. Each SON 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 SON peer can offer the same services subset at the same domain and publish it on the same aggregation peer. In the interest of simplification, the search concludes with the first match though AgS has the ability to return all matches.

Each execution simulates 50 hours of work. For each AgS size, each simulation used the same scenario with a different number of search operations, which ranged from 100 to 1000 operations step by 100. For each AgS size, the plotted value is the average of these 10 simulations. In order to optimize the searching process, caching of the search results was also taken into account. Two sets of simulations were executed comprising two scenarios involving the P2P SON: 1) with the AgS and 2) without the AgS. A third scenario regarding the P2P SON with Gnutella was also executed. The PeerFactSim.KOM [12] discrete events simulator was used in the simulations.

A. Results

The values for the RT metric were obtained by measuring the time a Query Message takes to reach the aggregation peer that has the desired service descriptor. Each peer in the travel path forwards the Query Message in the case the service descriptor is absent in its local cache. When the search operation starts, the Query Message receives a time stamp (TS) that is used at the arrival of the corresponding Query Reply Message to calculate the response time. Thus, RT will be the ratio between the accumulated time for all successfully accomplished search messages and the number of these messages. Eq. 1 shows the formula for RT. It is worth mentioning that results rely on a confidence interval (C.I.) of 95% for the analyzed average values, and curves are drawn with the margin of error considering that C.I.

$$RT = \frac{\sum_{t=0}^{n_{infoFound}} (CurrentTime - TS_t)}{QueryReplyMessages} \quad (1)$$

A first important finding is that the search hit ratio of the AgS that is almost 100%. Despite the fact that AgS is an unstructured overlay, it guarantees the finding of the information in case the information is published in it. The search hit ratio is very similar for the environment without AgS. This is explained by the utilization of the same overlay at the P2P SON.

Fig. 2 shows the comparison between the environments with and without AgS regarding RT. Fig. 2 is quite revealing in several ways. First, it shows the response time is lower when using AgS. It is possible to note that RT remains lower than 500ms. Comparing the arithmetic mean of the plotted values it is possible to figure out that RT using AgS is around 53% lower than in the P2P SON environment. The AgS scalability is also apparent in Fig. 2 since while the increasing factor for aggregation peers is 100, the increasing factor for RT is around 8.5. This means that the response time increases slower than the number of aggregation peers, i.e., service provider can use bigger AgS in order to reach more users with relatively minimal increase in response time for searching their services and component services.

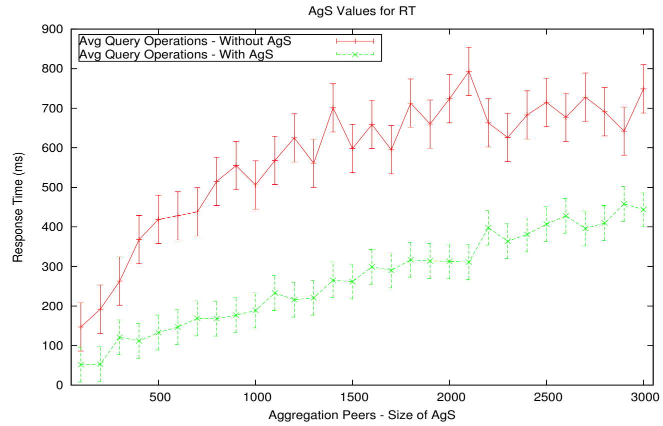

Fig. 2. Compared response time - with and without AgS

B. Performance Comparison

Simulations using the same scenarios and number of query operations were executed using Gnutella 6.0 in order to compare the AgS performance.

Fig. 3 presents the plotted averaged RT values for the environment using AgS and for the environment using Gnutella. It is possible to note the curve representing Gnutella presents lower response time than the one representing AgS, except for small sized overlays. However, in these environments composed of little number of nodes, the environment with AgS presented better or around similar performance. Thus, comparing the average values of these two curves, the environment with Gnutella will be around 48% lower than in the environment with AgS. This difference advocates the use of Gnutella concerning the searching and might represent a drawback for AgS. However, the searching performance is not the only characteristic a provider should take into account when choosing the environment where to offer their services.

The search hit ratio provides clues about the search efficiency in terms of started and successful concluded queries. As already mentioned the search hit ratio for the environments with AgS and without AgS is almost 100%. Thus, although the AgS response time is not as good as in Gnutella, the AgS search hit ratio is better.

The simulations on the scenario with Gnutella reveal that on average the search hit ratio for the environment with Gnutella is around 69%. This value is considerable lower than the 98% achieved with AgS. Taking into account the averaged values for the search hit ratio in AgS and Gnutella environments, the positive bias in favor of AgS is around 42%.

An important finding can be made crossing the data of search hit ratio and performance on Fig. 3. It is possible to realize that for small overlays (maximum of 600 nodes), the AgS search hit ratio as well as the searching performance are better than in the Gnutella environment. Taking into account this finding, it is possible to claim the AgS presents better conditions to satisfy consortiums of service providers that use a small number of nodes to form their P2P SON in order

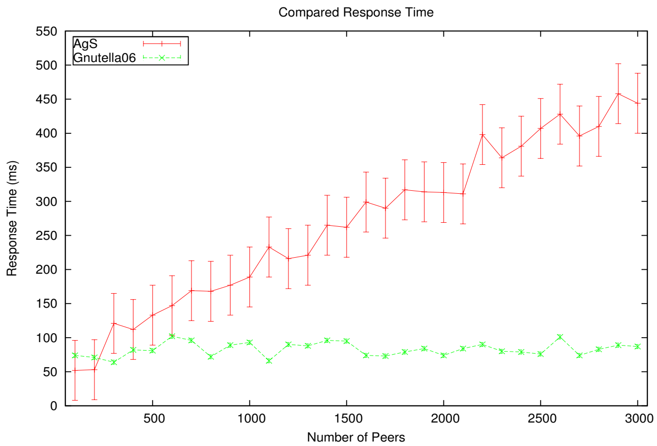

Fig. 3. Compared response time - AgS and Gnutella

to share the services offering infrastructure. This can be the case, for instance, for specialized service providers that target a specialized market niche such as the subtitling and dubbing video content market.

V. Conclusion

This paper addressed a performance evaluation in the context of the Aggregation Service (AgS). The AgS is a multi-domain service and service components searching optimization mechanism based on P2P. It relies on the publication of the service offerings in a second-tier P2P overlay, which optimizes the searches. To accomplish the evaluation, the AgS was explained and simulated. The simulation took into account the response time (RT) as assessment metric. The performance of AgS was also compared with the performance of a scenario using Gnutella as the searching mechanism.

The preliminary results show that the AgS reduces the response time about by 53% when compared with the same query operations in a scenario that does not use AgS. Even with this reduction, the AgS search hit ratio, which is around 98%, remains unaltered. Results also show the AgS scalability is good. Although there is a direct relation between the number of nodes and response time, this rate is low. For AgS, while the increasing factor for aggregation peers is 100, the increasing factor for RT is around 8.5 (i.e. around 11:1).

Preliminary results comparing AgS and Gnutella show a trade off. Despite the fact that Gnutella is faster, its search hit ratio is approximately 42% lower than AgS. On the other hand, AgS is around 48% slower. Nonetheless, for small environments (maximum of 600 nodes) the bias is positive for AgS, independently of response time or search hit ratio. This finding leads to the conclusion that AgS is well suited for specialized market niches where the overlays formed by the active service providers are small.

Further studies are still necessary. The maintenance of the AgS data consistency is an open issue. 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 P2P SON peer found by the query operation in order to verify the component service availability. This approach may prevent the overhead of the P2P SON peers' polling monitoring messages. The balance between performance and the minimal number of aggregation peers in AgS is another planned contribution. An approach to address that is the fine-tuning of the parameters that control AgS. Interested service providers can use these expected results to plan their services offering infrastructure in order to keep costs under control.

Planned future work also comprises the assessment of the search efficiency in terms of the percentage of P2P SON peers that form the AgS. This assessment should take into account the searching response time and the overhead associated with the AgS maintenance operations (e.g., nodes joining, services publishing, consistency checking, nodes leaving, etc.).

Acknowledgment

This work was partially funded by Portuguese Foundation for Science and Technology (grant SFRH/BD/45683/2008).

References

[1] C. Tran and Z. Dziong, "Service overlay network capacity adaptation for profit maximization," *IEEE Transactions on Network and Service Management*, vol. 7, no. 2, pp. 72–82, Jun. 2010.

[2] A. Fiorese, P. Simões, and F. Boavida, "Service searching based on P2P aggregation," in *The 24th International Conference on Information Networking (ICOIN 2010)*, Busan, South Korea, 2010.

[3] ——, "An aggregation scheme for the optimisation of service search in Peer-to-Peer overlays," in *Proceedings of the 6th International Conference on Network and Services Management (CNSM 2010)*, Niagara Falls, Canada, Oct. 2010.

[4] R. L. Vianna, M. J. B. Almeida, L. M. R. Tarouco, and L. Z. Granville, "Investigating web services composition applied to network management," in *Proceedings of IEEE International Conference on Web Services*. Chicago, USA: IEEE Computer Society, 2006, pp. 531–540.

[5] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity search for web services," in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*. Toronto, Canada: VLDB Endowment, 2004, pp. 372–383.

[6] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, and C. Zimmer, "Discovering and exploiting keyword and attribute-value co-occurrences to improve P2P routing indices," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. Arlington, Virginia, USA: ACM, 2006, pp. 172–181.

[7] H. Shen, "A P2P-based intelligent resource discovery mechanism in internet-based distributed systems," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 197–209, Feb. 2009.

[8] P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi, "Peer-to-Peer resource discovery in grids: Models and systems," *Future Gener. Comput. Syst.*, vol. 23, no. 7, pp. 864–878, 2007.

[9] A. Brocco, A. Malatras, and B. Hirsbrunner, "Proactive information caching for efficient resource discovery in a self-structured grid," in *Proceedings of the 2009 workshop on Bio-inspired algorithms for distributed systems*. Barcelona, Spain: ACM, 2009, pp. 11–18.

[10] N. Limam, J. Ziembicki, R. Ahmed, Y. Iraqi, D. T. Li, R. Boutaba, and F. Cuervo, "OSDA: open service discovery architecture for efficient cross-domain service provisioning," *Computer Communications*, vol. 30, no. 3, pp. 546–563, 2007.

[11] E. Lavinal, N. Simoni, M. Song, and B. Mathieu, "A next-generation service overlay architecture," *Annals of Telecommunications*, vol. 64, no. 3, pp. 175–185, Apr. 2009.

[12] A. Kovacevic, S. Kaune, N. Liebau, R. Steinmetz, and P. Mukherjee, "Benchmarking platform for peer-to-peer systems," *it - Information Technology*, vol. 49, no. 5, pp. 312–319, Sep. 2007.