# SEMANTIC RESOURCE DISCOVERY IN GRID AND MULTI-AGENT ENVIRONMENT

Muntasir Al-Asfoor and Maria Fasli
*School of Computer Science and Electronic Engineering, Essex University, Wivenhoe Park, Colchester, U.K.*

Keywords: Grid, Resource Discovery, Semantic Matching, Network Performance.

Abstract: Resources sharing has become an evolved field of study for the distributed systems communities. Enabling geographically diverse computational entities to share resource in a seamless way regardless of the hardware and software specifications has become a need by researchers communities. Resource discovery plays a vital role in the sharing lifetime. Resource sharing has been studied in this paper as a network activity. The effect of the locations where the semantic matching is done on the network performance has been investigated. An experiment has been designed to implement the proposed scenarios in a simulated environment. As part of this experiment a semantic matching algorithm based on reference ontology has been also implemented. The experimental results have demonstrated that doing a matching process in the requesters nodes is less network time consuming, giving that the requester has a copy of the neighbors resources descriptions.

## 1 INTRODUCTION

The last decade has shown a rapid increase in the researchs projects which focus on distributed computing. One of the most interested focuses was on the development of feasible techniques that allow distributed entities to share resources. Analogous to an electricity power grid, Grid computing views computing, storage, data sets, expensive scientific instruments and so on as utilities to be delivered over the Internet seamlessly, transparently and dynamically as and when needed, by the virtualization of these resources (Ludwig and Santen, 2002) and (Freeman et al., 2006) One of the key issues of resources sharing is the process of resources discovery; it defines the process of locating the available resources or services and retrieving their descriptions (Timm, 2005). Nowadays, the syntax based and name lookup matching techniques used by web search engines lack the ability to discover a service or resource according to the meaning of the term that represents them. They employ a simple string matching to compare two terms with only two possibilities: either finding the exact match or not. Accordingly, these techniques are not suitable for Grid or distributed environments where different users might describe the same term in a different way. Semantic matching by employing ontologies has become a feasible solution to overcome the syntactic matching problems. As resource sharing across distributed systems is a network activity we have focused in this paper on the effects of the matching process and the location of the nodes which do the matching on the network performance. A simulation tools have been used to evaluate the proposed scenarios and to measure the system performance in terms of delay time and network throughputs. The rest of this paper has been organised as the follows: section 2 has been devoted to the related work in the field of resources sharing in the distributed systems more precisely Grid environment. The proposed scenarios have been discussed in details in section 3. Section 4 shows the experimental results and system evaluation in a simulation environment. The last section has been dedicated to the conclusions and our view of the possible future development to enhance the system performance.

## 2 RELATED WORK

Computational resources sharing has been an evolved topic of research in both academia and industry. Researchers and developers in the fields of networking and distributed systems have proposed many frameworks which manage and monitor resources advertisement and discovery process. The Globus toolkit (Schopf et al., 2006) Monitoring and Discovery System (MDS4) has developed a discovery techniques for


Al-Asfoor M. and Fasli M..
SEMANTIC RESOURCE DISCOVERY IN GRID AND MULTI-AGENT ENVIRONMENT.
DOI: 10.5220/0003750803660370
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 366-370
ISBN: 978-989-8425-96-6

a distributed systems based on WSRF (Web Services Resource Framework). The centralised management approach has made MDS4 fragile to the problem of halting the nodes which behold the registration information. Different techniques were used by (Han and Berry, 2008) for semantic discovery in a Grid environment. They considered the system with super nodes that hold resources. Users can locate a resource by performing a desired web service query. The system can help the user to search the web services which match his requirement and then notify that user. The paper uses Profile matchmaking techniques to decide the degree of matching two concepts. (Castano et al., 2003) have proposed an algorithm for resource discovery based on the idea of considering both linguistics features of the concepts in the ontology as well the semantic relations among concepts in a peer ontology. They made use of the H-MATCH algorithm to compute the degree of similarity between two terms represent two concepts. The first step in this algorithm is to use the WordNet thesaurus paths to compute the Linguistic Affinity (LA). Secondly, they compute the Relational Affinity (RL) for the concepts relations and properties based on weights taken from the ARTEMIS (Tuchinda et al., 2004) framework. In contrast with what we have mentioned above, our concern is to the resource sharing problem as a network activity with the aim of improving the network functionality. The work present focuses on network aspects of the system like (network topology, network technology) with the aim of improving factors like (delay, throughput, etc.).

# 3 THE PROPOSED SCENARIOS

During the course of this work two main resource discovery activities have been studied. The first is the matching process between the request and the advertisements; the second is the interaction among the requester nodes and the resource providers nodes. With the aim of studying the effect of each of these activities on the other, this research has investigated the impacts of doing the matching process in the requester node and compared it to the standard way of doing it at the providers nodes in terms of the network quality of services and the request satisfaction time. Accordingly, two scenarios have been engineered to study the two cases as shown in subsections 3.1 and 3.2.

## 3.1 The First Scenario: the Requester does the Matching Process

In the first scenario, the system has been engineered in a way that enables the requester node to do the semantic matching process and decide which provider to contact. In this scenario we have assumed that the requester had been informed about the available resources in its neighbors. A flow diagram of this scenario is shown in Fig.1

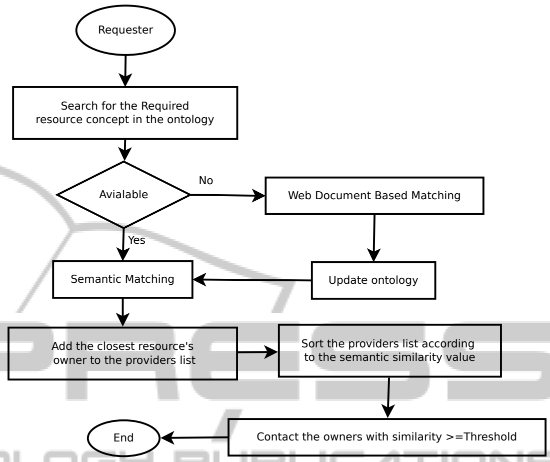


Figure 1: The first scenario data flow diagram.

An ontology based semantic matching method has been used that enables to the node to match the resources with a standard tree-like structured reference ontology[1] using the semantic matching algorithm proposed by (Ge and Qiu, 2008). Based on this method, the requester finds the best match by computing the semantic distance among concepts.

The semantic distance has been computing based on the summation of the weights which has been assigned for each edge connecting two concepts in the shortest path between the two concepts subject to semantic matching as shown in Equation 1. The weights are computed by using Equation 2. which assigns weights according to the position of the concepts in the ontology. The purpose of edge weights is to distinguish between the more and less general concepts.

$$SD = \sum_i W(c_i), \qquad c_i \in SP \tag{1}$$

Where: $SD$: is the Semantic distance. $SP$: is the shortest Path between the two concepts subject to matching.

$$W(c) = 1 + \frac{1}{2^D} \tag{2}$$

Where: $W(c)$ : is the weight of the concept $c$, $D$: is

[1] A simple computer ontology has been used for simulation purposes. For instance, the computer has hardware and software components, then the hardware consists of CPU, Memory, Hard Disk, etc.

the Depth (level) of the concept $c$ in the reference ontology.

Subsequently, the semantic similarity is computed using the hypothesis that greater semantic distance between concepts means smaller similarity and vice versa (Roelleke and Wang, 2008). As the semantic distance could range widely in a non-normalised way, then there is a need for a normalization function to convert the semantic distance values in to a logically acceptable semantic similarity values which could then be used to rank the resources providers as shown in figure 1. To do so, and as describes in (Ge and Qiu, 2008), the semantic function needs to support three main properties:

- The semantic similarity values are a real numbers in the range between [0, 1].
- The semantic similarity between any concepts and itself = 1.
- he relation between semantic distance and semantic similarity is inversed.

Accordingly, for the purpose of this paper we have used a linear function to compute the semantic similarity from the semantic distance as proposed by (Ge and Qiu, 2008) which is shown in Equation 3.

$$SSem = \frac{1}{(p + SD + 1)} \qquad (0 < p \leq 1) \tag{3}$$

Where: $SSem$ is the semantic similarity value.

From the experimental results we have chosen $p$ which produced the best subjectively observed results. After that, the requester will sort the providers based on the semantic similarity value and contact the ones with values  a predefined threshold.

Using a reference ontology during the matching process has raised the problem of the concept is not part of the ontology; to overcome this problem we have used a web document based matching (WDM) technique to find the closest existing concept. Using this technique, the system fetches context information related to the concept from web sites like Wikipedia[2] then applies TF-IDF (Term Frequency-Inverse Documents Frequency) algorithm (Roelleke and Wang, 2008) to find the closest existing concept in the ontology. Afterwards, the system applies the same matching steps on the existing founded concept.

In this scenario, there are two types of delay time: the first one is the time required to compute the similarity and rank the resource providers accordingly; this time has no effect on the network traffic since its done locally in the node.

[2]See http://en.wikipedia.org.

The other time is the one required to contact the providers and receive the acknowledgements. Since the requester has done the matching process locally it does not need to send large messages to contact the providers, it should be just small messages to insure that the resource is still available and the provider still happy to share it.

In terms of request satisfaction time which is the summation of all the delay times from the beginning of the matching process until receiving the acknowledgment both times have impacts on it. As the main argument of this research is to study the effects of the place where the matching is done on the network performance as well as the request satisfaction time; we have developed another scenario where the requester sends a request to its neighbours and the neighbours themselves do the matching process individually, this scenario is shown in the next section.

### 3.2 The Second Scenario: the Providers do the Matching Process

Using this topology, there is no need for the nodes to have the others resources information but they must have the reference ontology to perform the semantic matching base on it. In this case, the same matching steps discussed in the first scenario have to be done but in the providers nodes. The requesters node has to send a request contains the resource description to its neighbours and wait for reply. On the other part, the provider receives the request(s) and does the matching steps on its own resources database and return the highest available resource semantic similarity value. Accordingly, the requester will collect the replies and rank the providers to be contacted based on the semantic similarity values provided by the providers.

Using this scenario, the process of resource discovery will affect the network traffic as more messages need to be broad-casted. Furthermore, the requester needs to wait the providers to finish the matching process and return the result. In addition, one provider would receive many requests, which leads to the request to be queued in the providers node for the previous ones to be done. For a network point of view the total network delay time could be calculated using Equation 4.

$$T = R_t + q_t + mp_t + Ack_t \tag{4}$$

Where: $T$: is the Total network delay time, $R_t$: is the time required to send the Request from the requester to the provider. $q_t$: is the queue time in the providers side. $mp_t$: is the matching time, $Ack_t$ is the time required to send an acknowledgement back to the requester.

The same equation could be used to compute the request satisfaction time as the requester has to wait for $T$ time to accomplish its job. For the first scenario, the network time could be computed using the same equation but after removing the matching time..

# 4 SYSTEM EVALUATION AND EXPERIMENTAL RESULTS

As the proposed system consists of two different parts (matching and networking), the evaluation process has been divided in to two parts. The first part has been devoted to test the matching algorithm being used in the system. As mentioned before, the matching process based on reference ontology and the concepts being matched structurally using a semantic distances computation. Accordingly, Equation 3 has been used to compute the semantic similarity among concepts through the ontology as shown in table 1. The results show the semantic values started with 1 (two concepts are exactly the same) and the decreased gradually as the distance between the concepts increased through the ontology. For example, the concepts XP has a semantic value of (1) with itself and then (0.81) with Windows which is its super concept and the value decreased according to the target concept location in the ontology.

As the main objective of this paper is to study the system performance from a network point of view, an experiment has been conducted for this purpose. Using the network simulator (NS2)[3]; the basics of the experiment is to compare between two cases: case1 where the system performs the matching process locally in the requester node giving that the requester has a copy of the neighbours resources databases. In this case, the node which needs some extra resources has to check the availability of the required resources in the neighbour nodes. The assumption in here is the nodes had received as advertisements the neighbours resources information and stored them locally. Accordingly, the node should not send requests until knowing which neighbours have the required resources, at least during the time of advertisements. In this situation, the main part of resource discovery could be done locally without using any network resources (in this case network traffic time).

In case 2, the system does the opposite procedure; the node requires extra resource has to send requests to its neighbours and the neighbours shall do the matching and return the results. Using this technique, all the resource discovery process will be done

[3]See http://www.isi.edu/nsnam/ns.

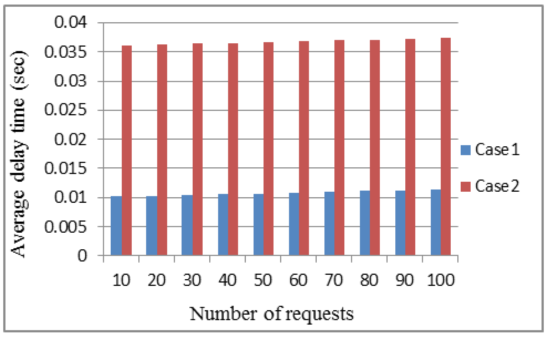


Figure 2: Average delay time comparison between case 1 and case2.

across the network (i.e. it involves more network traffic time). In contrast with case1, the requester has to wait for all the providers to do the matching to receive the results and then contact the best provider. To study these situation have simulated the two cases and run the simulation for different number of requests to evaluate the system performance (from a network point of view). The first parameter has been studied was the time delay from the requester sends request to the time it receives the results from the providers as shown in Fig. 2.

The average delay time has been slightly increased as the number of requests increased for both cases because more requests means more waiting time in the queue which leads to more traffic time. At the same time, the results show an noticeable difference between case1 and case2 in terms of delay time which means case2 needs more traffic time to satisfy the requests. Another performance measure has been used during the course of this work which is the system throughput (the number of bits received per second). In this paper we have used the positive throughput which takes in account only the data that correctly received per second. Fig. 3 shows the difference in system throughput between case1 and case2 for a variety number of requests. It shows clearly that the system throughput increases as the number of requests increases. The reason for this increase is the increase in delay time is slow in compare to the increase in the amount of date being sent. In case2 the situation is the opposite as the throughput increases dramatically as the number of request increases because more requests requires more matching time which increases the total delay time.

Table 1: The semantic values for a subset of the concepts .

| | Computer | CPU | Operating System | Memory | Windows | UNIX | XP | Windows 7 |
|---|---|---|---|---|---|---|---|---|
| Computer | 1 | 0.76 | 0.76 | 0.76 | 0.65 | 0.65 | 0.58 | 0.58 |
| CPU | 0.76 | 1 | 0.62 | 0.62 | 0.54 | 0.54 | 0.49 | 0.49 |
| Operating System | 0.76 | 0.62 | 1 | 0.62 | 0.81 | 0.81 | 0.70 | 0.70 |
| Memory | 0.76 | 0.76 | 0.76 | 1 | 0.65 | 0.65 | 0.58 | 0.58 |
| Windows | 0.65 | 0.54 | 0.76 | 0.54 | 1 | 0.68 | 0.85 | 0.85 |
| UNIX | 0.65 | 0.54 | 0.76 | 0.54 | 0.68 | 1 | 0.61 | 0.61 |
| XP | 0.58 | 0.49 | 0.70 | 0.49 | 0.81 | 0.61 | 1 | 0.76 |
| Windows 7 | 0.58 | 0.49 | 0.70 | 0.49 | 0.81 | 0.61 | 0.76 | 1 |


Figure 3: Throughputs (bits per second) comparison between case1 and case2.

## 5 CONCLUSIONS AND FUTURE WORK

The rapid increase in the number of applications has led to the need of more expensive resources to satisfy this growth. To cope with this dynamic growth, resources sharing has become a suitable solution where more resources could be shared as the system grows. Resources discovery plays the main role during the sharing life time. In this paper, we have proposed a resource discovery mechanism and studied two different scenarios to implement this mechanism. A semantic matching algorithm has been implemented and the system performance from a network point of view has been studied. The experimental results have demonstrated that doing the matching process in the requesters node would save time and increased the network throughput although the requester has to have the neighbours nodes resources information which means more storage/updating overhead. There are many possibilities for future work, the system could be improve by providing some sort of nodes/agents federation based on some classification criteria like geographic location, agents capabilities or the type of resources. This federation gives the nodes some knowledge about teach other which could enhance system performance and decrease the network, matching and storage overheads.

## REFERENCES

Castano, S., Ferrara, A., and Montanelli, S. (2003). H-match: an algorithm for dynamically matching ontologies in peer-based systems. In *In Proc. of the 1st Int. Workshop on Semantic Web and Databases (SWDB) at VLDB 2003*, pages 231–250.

Freeman, T., Keahey, K., Foster, I., Rana, A., Sotomoayor, B., and Wuerthwein, F. (2006). Division of labor: Tools for growing and scaling grids. In *of Lecture Notes in Computer Science*, pages 40–51. Springer.

Ge, J. and Qiu, Y. (2008). Concept similarity matching based on semantic distance. In *Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid*, SKG '08, pages 380–383, Washington, DC, USA. IEEE Computer Society.

Han, L. and Berry, D. (2008). Semantic-supported and agent-based decentralized grid resource discovery. *Future Gener. Comput. Syst.*, 24:806–812.

Ludwig, S. and Santen, P. V. (2002). A grid service discovery matchmaker based on ontology. In *In EuroWeb 2002. British Computer Society*, pages 17–18.

Roelleke, T. and Wang, J. (2008). Tf-idf uncovered: a study of theories and probabilities. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 435–442, New York, NY, USA. ACM.

Schopf, J. M., Pearlman, L., Miller, N., Kesselman, C., and Chervenak, A. (2006). Monitoring the grid with the globus toolkit mds4. *Journal of Physics: Conference Series*, 46.

Timm, I. J. (2005). Large scale multiagent simulation on the grid. In *Proceedings of 5 th IEEE International Symposium on Cluster Computing and the Grid. IEEE Computer Society*, pages 334–341.

Tuchinda, R., Thakkar, S., Gil, Y., and Deelman, E. (2004). Artemis: Integrating scientific data on the grid. In *AAAI*, pages 892–899.