A Distributed Ontology Framework in the Semantic Grid Environment

Andrew Flahive, Wenny Rahayu
La Trobe University, Australia
{A.Flahive, W.Rahayu}@latrobe.edu.au

David Taniar
Monash University, Australia
David.Taniar@infotech.monash.edu.au

Bernady Apduhan
Kyushu Sangyo University, Fukuoka, Japan
bob@is.kyusan-u.ac.jp

Abstract

This paper explores a distributed ontology framework for tailoring ontologies in the Semantic Grid environment. The framework is divided into five main categories; Ontology Processing, Ontology Location, Ontology Connection, Users' Connection and Algorithm Location. A number of possible scenarios are discussed following two Case Studies that indicate how the framework can be manipulated and used in various situations. This framework will help developers design tools for tailoring ontologies in the Semantic Grid Environment.

1. Introduction

The next generation of the Internet is called the *Semantic Web*. It will provide an environment that allows more intelligent knowledge management and data mining. The main focus is the increase in formal structures used on the Internet. The taxonomies for these structures are called ontologies [5, 6], and the success of the semantic web highly depends on the success of these ontologies.

The Semantic Web is becoming popular for sharing information and resources over the Internet through *Web Services*. These Web Services are being developed, for the Semantic Grid, in an attempt to decentralize systems and share computing ability. Both hardware and software resources are being integrated into the sharing domain.

The *Semantic Grid* is similar to the Semantic Web, with the only difference being that the former shares resources in accordance to certain architectures and standard grid infrastructures[4, 7]. The standards adhered to in the Semantic Grid allows anyone around the world to develop their own tools to interact with semantic grid resources easily. Virtual organizations that allow access to large computing resources are becoming increasingly popular.

In this fashion, communities are starting to share large ontologies in the Semantic Grid Environment as it is an ideal medium for such a knowledge resource. But often these ontologies are too cumbersome for any individual to use it in it's entirety. Currently Grid resources are being developed to help users tailor these large ontologies[2, 1] into a more user friendly form.

This paper discusses the framework for developing an ontology tailoring system for the Semantic Grid Environment. Five Categories make up all of the possible scenarios for this framework and their implications are discussed in this paper. Determining which options in the categories should be chosen, and why, is a case by case problem. Two Case Studies discuss how to manipulate the Distributed Ontology Framework when some user constraints have been applied. This paper aims to help the system developer tune their system to make ontology tailoring more time efficient and cost effective for the intended user.

2. Background

Ontologies may be tiny, containing just a few concepts and relationships or they may be ever expanding, containing many millions of concepts and relationships. Ontologies are becoming popular largely due to what they promise: a shared and common understanding of a domain that can be communicated easily between people and applications.

Ontologies can represent many domains of knowledge whilst being machine understandable. However, traversing large ontologies and fulfilling specific user demands, often takes many computer hours to complete.

Deriving Tailored Ontologies from large base ontologies enables individuals to use only specific parts of the ontology for their daily use. Most user applications only require particular aspects of the ontology as they do not benefit from the plethora of semantic information that may be present in the ontology.

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

Ontology tailoring, is computationally expensive, in part because of the size of the ontologies and also partly because of the complexity of the requirements of the user. By developing an ***O**ntology **T**ailoring **P**rogram **R**esource* (OTPR) for use on the Semantic Grid many people may be able to process their computationally expensive tasks cheaply. It will allow people from remote locations [3] to tailor ontologies from large base ontologies without requiring their own High Powered Computing (HPC) facility [8].

3. Distributed Ontology Framework

Figure 1 shows an overview of the proposed distributed ontology framework in the Grid Environment. This type of framework was briefly introduced in the paper [2]. This paper builds on that initial idea to by discussing some guidelines to the categories to show how the framework should be used.

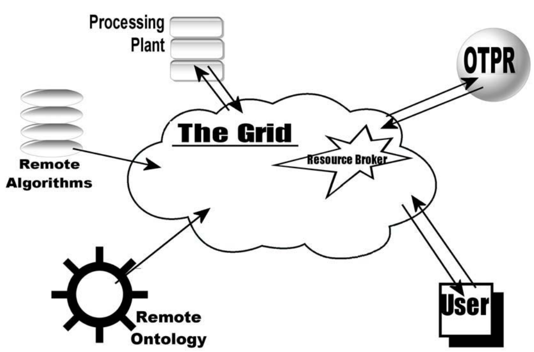

Figure 1. Components of the Framework in the Grid Environment

Basically the OTPR is a Resource in the Semantic Grid that enables users to connect to it so that they can perform their own ontology tailoring. The user submits a base ontology along with their specifications (algorithms) to the OTPR. The OTPR will either process the work itself or use a another resource on the Grid to complete the work. Once the tailored ontology has been derived the results are passed back to the user.

The framework is broken down into five basic categories; (i) Ontology Processing, (ii) Ontology Location, (iii) Ontology Connection, (iv) User Connection and (v) Algorithm Location. These categories make up the different variations of the distributed ontology framework [2].

3.1. Ontology Processing

The OTPR can either process the ontology itself or use a processing plant (a cluster of processing units) to do the bulk of the work for it. A processing plant is required in some situations because of the sheer amount of computations that need to take place in order manipulate the ontology more efficiently. Whereas a single processor machine may finish the task in a few hours, a processing plant will be able to complete the same work in only a few minutes or even seconds.

Column 1 in Table 1 indicates when it is most appropriate to use the processing plant and when it would be better to use OTPR's own processing power. If the OTPR can't process the work itself it must chose to either connect directly to a processing plant (if one is known) or use a resource broker to find a suitable one.

3.2. Ontology Location

Ontologies can located in either of two main locations, at the User's location or at a remote location. Column 2 in Table 1 indicates where it is best to store the Ontology given certain constraints. If the user's connection is slow, it would be best to house the Ontology at a remote location. However, if the user has a fast connection to the OTPR, then, rather than risking a slow remote connection, it would be best to store the ontology at the user's site.

3.3. Ontology Connection

The OTPR requires some sort of access to the Ontologies so that it can perform the required tasks. The OTPR can either copy the Ontology up to a local storage space or use special commands to request and traverse the ontology remotely. Column 3 in Table 1 indicates when it is best to copy up the ontology or have to access it remotely through a set of commands. If the user only has a slow connection to the Grid then copying up the whole ontology may take too long, no matter how large it is. On the other hand, if they have a fast connection, copying the ontology will greatly increase the speed of the overall ontology tailoring process.

3.4. Users Connection

Is the address of the OTPR known to the user or does the user require a resource broker to match them up with the OTPR? Column 4 in Table 1 indicates how the user should connect to the OTPR. If the User has a slow connection to the OTPR, then it may be best for them to submit their information to the resource broker and let them deal with the OTPR requests.

	Ontology Processing	Ontology Location	Ontology Connection	User Connection	Algorithm Location
A Large Ontology	Processing Plant, Direct Connection	N/A	Store at the OTPR	N/A	N/A
A Small Ontology	at the OTPR	N/A	Either store at Remote or Local	N/A	N/A
Slow User Connection	N/A	At the Remote Location	N/A	Through Resource Broker	at the ORPR and/or remote loc.
Fast User Connection	N/A	at the User	Store at the OTPR	Connect Directly to OTPR	At the User and/or OTPR
Large amount of Processing	Processing Plant, Direct Connection	N/A	Store at the OTPR	N/A	N/A
Small amount of Processing	at the OTPR	N/A	Connect to Remote Location	N/A	N/A

Table 1. How to optimize your system to take advantage of the Semantic Grid Environment

3.5. Algorithm Location

Algorithms cab be used to tell the OTPR how the ontology should be manipulated to get the required view of the large ontology. Algorithms from many sources can be combined. It would not be uncommon for the user to require OTPR's generic algorithms, a remote set of algorithms and their own developed algorithms to process the ontology.

The last column in Table 1 shows which locations are preferred over others given certain conditions. However, the algorithms are generally small in size compared with that of the ontology and their location before processing will have little impact on the overall processing time.

4. Case Studies

By combining one or more of the possibilities from each of the above categories, all of the variables of a particular system are captured. To enable a better understanding, two case studies are discussed below and deduce the best framework for certain scenarios.

4.1. Scenario 1.

The Problem: The user requires an ontology to be processed in the shortest time possible. The ontology is large and many algorithms need to be applied. The network speeds of this particular grid are fast. What is the best framework so that the ontologies can be processed quickly?

The Solution: The user should connect directly to the OTPR as this will cut out any unnecessary network traffic. The ontology should be located with the user and then copied up to the OTPR's local storage space. This will allow the large ontology to be processed more efficiently, as many calls to the ontology may be needed due to it's size and the number of algorithms needing to be applied. The algorithms required can be kept with the user to avoid any unnecessary network traffic but not necessary.

Due to the large amount of processing required, the OTPR should employ a known supercomputer to process the work. Using a resource broker will increase network traffic as well as increase the risk of employing a slower processing plant. Cost here, isn't a factor, so no expense should be spared in securing the best parallel processing system for the job. This framework will give the user the best chance in getting their ontology processed in the quickest time possible.

4.2. Scenario 2.

The Problem: A large ontology has to be processed but the time frame is relaxed. Network speeds are poor around this grid, even the user has an unreliable connection. Should the ontology be copied up to the OTPR or would it be more efficient to store the Ontology at the Users site?

The Solution: It is assumed that the user wants to spend as little money as possible. It would be best if the Ontology were processed at a Processing Plant as processing large on-

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

tologies on a single processor machine may take even longer than the relaxed time frame that the user has imposed. The OTPR should connect through a resource broker so as to find the cheapest processing plant that is still capable of doing the job. The Algorithms should be located together at any one site. The user may like to connect through a resource broker to find a suitable OTPR, one that perhaps isn't too expensive.

The ontology may be located at either the users site or a remote site. The ontology should however, be copied up to the OTPR's local storage space. This should be done for two main reasons. The first is that an open connection between the remote site and the OTPR will need to be maintained for a long period of time. On a network that may drop out it would be better to copy the ontology up, close the connection and wait for a response from the OTPR when the job is done. The second is because of the large number of expected accesses to the ontology may degrade the network even further if it were to be located remotely from the OTPR for the duration of the process.

5. Discussion

In order to define the right framework to match a certain situation, a number of factors must be taken into account. These factors play a vital part and some of these are discussed below.

The Size of the Ontology plays a vital part in the decision process. Will the ontology take too long for the OTPR to copy to it's local storage space? Can the OTPR traverse such a large Ontology efficiently on its own or will it require some sort of Processing Plant? Are any **Time Constraints** imposed by the user? Does the OTPR have enough time to process the ontology given the user time constraints? Will a processing plant be required? **The Demand on the OTPR** also determines what priority is given to efficient processing. If the OTPR is in high demand then it is in the OTPR's best interest to process as many jobs as quick as possible.

The User's Bandwidth also plays a part. If the user has a slow connection to the OTPR, then providing the OTPR with a link to the location of the remote Ontology would be better than storing the ontology at the user's location.

The **Algorithm Complexity** plays a major role in deciding whether or not the OTPR should request the use of a Processing Plant, especially when the algorithms are working on large ontologies. The **Number of Algorithms** also needs to be evaluated. Fewer algorithms may mean that less work has to be done.

The **Cost** of employing a Processing Plant and the **Resource Broker Cost** are both major factors that need to be analyzed thoroughly. The time constraints imposed by the User may have to be reconsidered if keeping costs down is a higher priority.

By thoroughly discussing all of these factors, the system developer may narrow the number of possible scenarios down to just a few. Choosing the right scenario is important, as it will save time and money. They system developer should then manipulate the framework provided to suit their own requirements.

6. Conclusion

This paper discussed a distributed framework for tailoring ontologies efficiently in the Grid Environment. It was found that these five categories combine to form many scenarios that the system developer must take into account when proposing such a project.

The two case studies presented some common issues that developers will need to consider. For a tight time constraint and large budget, the fastest super computer available should be hired. For a more relaxed time constraint, a resource broker should be used to find a capable but cheap machine for the job. This paper discussed a number of issues affect how the distributed framework can be manipulated to be assured of the best solution. Future work may involve defining issues more precisely with an aim to produce an expert system to help solve most case scenarios.

References

[1] Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., and Taniar, D. A distributed approach to sub-ontology extraction. In *18th Int'l Conf. on Advanced Information Networking and Applications (AINA'04)*, Japan, 2004.

[2] Flahive, A., Rahayu, W., Taniar, D. and Apduhan, B. A distributed ontology framework for the grid. In *The 5th Int'l Conf. on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, Singapore, 2004. Springer-Verlag Heidelberg.

[3] I. Foster and C. Kesselman. *The Grid : Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 1999.

[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration, 2002.

[5] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, 1993.

[6] N. Guarino, M. Carrara, and P. Giaretta. An ontology of meta-level categories. In *KR'94: Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1994.

[7] ogsi wg. Open grid services infrastructure (ogsi) version 1.0. Global Grid Forum, version 1.0 edition, 2003.

[8] Plateau, B. The grid: Challenges and research issues. In *7th Asian Computing Science Conference*. Springer-Verlag Heidelberg, 2002.