Proceedings of the 18th IASTED International Conference
Parallel and Distributed Computing and Systems
November 13–15, 2006, Dallas, TX, USA
ISBN Hardcopy: 0-88986-638-4 / CD: 0-88986-640-6

GODIS: ONTOLOGY-BASED RESOURCE DISCOVERY AND INTEGRATION IN GRIDS

Juan Li, Iulian Radu, Son T. Vuong

University of British Columbia
2366 Main Mall, Vancouver, BC. V6T 1Z4
Canada
{juanli, vuong}@cs.ubc.ca; iulian@interchange.ubc.ca

ABSTRACT
This paper presents GODIS (Grid Ontological Directory and Integration System) a comprehensive architecture for resource sharing and discovery in large-scale grids, where nodes integrate local ontologies to expand semantic knowledge of shared grid resources. A peer-to-peer based DHT overlay is used to facilitate the formation of semantic communities on a large scale. Inside communities, ontological knowledge is integrated between nodes, and a DHT overlay is used for knowledge dissemination and discovery. This framework architecture improves interoperability among grid participants and aids efficient resource discovery through an expressive query language.

KEY WORDS
Grid, Semantic Web, Resource Discovery, and P2P

1. Introduction

The continued advances in networking technology and computer hardware have enabled more and more inexpensive PCs and various devices at the edge of the network to join the grid computing platform, therefore, the grid is shifting its focus from scientific computing to a pervasive, world wide resource sharing infrastructure. This large grid enables the sharing of a wide variety of resources, including hardware components, software packages, knowledge information, devices, and other grid services. However, the resource discovery problem in large-scale grids can be very challenging because of the abundance and the dynamic, heterogeneous and distributed nature of resources. Equally challenging is the task of integrating information about different properties and usage policies of these resources.

This paper focuses on the design of GODIS (Grid Ontological Directory and Integration System), an effective resource discovery and integration framework which copes with the complexity of Internet-scale grids. We describe appropriate enhancements for grid resource discovery, with an emphasized focus on integration and navigation of knowledge information. A semantic framework is proposed to increase the expressiveness and interoperability of grid resource discovery. This approach relies on ontologies to describe the structure and semantics of resource properties.

To improve scalability of this approach, the theory of "small-worlds" [1] is implemented to help grid nodes cluster into ontological P2P communities. Virtual communities are dynamically formed by reconfiguring the network with respect to shared ontological domains. The community discovery, construction, and maintenance are manipulated in a decentralized and automatic manner. Virtual communities help discriminatively distribute resource information and queries to ontologically related nodes, thus reducing the search space and improving the scalability. It is infeasible to expect a standard uniform ontology inside each community; instead, different nodes are allowed to contribute different ontologies focused on the same knowledge domain. An ontology integration strategy is presented which permits different community members to establish mappings among the various ontologies. Knowledge and resources within a community are accessible through an expressive ontology-based query language.

In the rest of the paper we detail the framework of the GODIS system. Section 2 describes related work. Section 3 details the two layers of our framework: directory overlay and community overlay. Section 4 explains knowledge integration, management and search in communities. Section 5 concludes with a summary of our work.

2. Related Work

In grids, resource discovery is usually guided by centralized servers. For example, in Globus Toolkit [2], users can get a node's resource information either by directly querying a server application running on the specific node, or querying dedicated information servers that retrieve and publish resource information of the organization. Techniques of how to associate information servers and construct an efficient, scalable network of directory servers are not specified.

To discover resources in more dynamic, large-scale, and distributed grid environments, P2P techniques have been used (e.g., [14, 15]). Flooding is the predominant search method in unstructured P2P networks. This method, though simple, does not scale well in terms of message overhead. An efficient approach is the use of distributed hash tables (DHTs) [3-6], which has been shown to be scalable and efficient. However, a missing feature is the inherent support for expressive queries.

Research on the Semantic Web project [7] has recently gained much attention for its knowledge integration vision. Its focus is to exploit the power of semantic technologies to aid in information representation, retrieval and aggregation over the web. Most of the Semantic Web projects use the standard RDF language [12, 13] to describe data. Ontology languages such as DAML+OIL [17] and OWL [18] build on top of RDF allow a user to describe relations between resources, defining a more abstract and expressive resource sharing environment.

In the last few years, several projects aimed to combine the strengths of grid and semantic web technologies, particularly for the use of resource sharing. For example, the UK myGrid [8] project uses ontologies to describe and select web-based services used in the Life Sciences, while OntoGrid [9] mixes in techniques from agent computing and peer-to-peer for distributed discovery semantic knowledge.

The combination of Semantic Web and P2P technologies can also serve to benefit each other. For example, in the InfoQuilt [10] system, nodes use the DAML+OIL ontological language to describe their information. Ontology knowledge is registered to a central server for efficient indexing, while queries are forwarded between nodes in a P2P manner. Helios [11] is another system that uses the P2P model for semantic knowledge sharing. Unlike InfoQuilt, both the ontology knowledge and data instance discovery are based on an unstructured P2P network. Our work differs from these approaches in the two-tier framework structure and ontology integration and navigation strategies.

3. Framework

3.1. Semantic Small-World

In social networks, a so-called small-world [1] structure has been observed. A small-world describes a loosely connected set of highly connected sub graphs formed around common data interests. The small-world networks exhibit special properties, namely, a small average diameter and a high degree of clustering, which make them effective and efficient in terms of distributing and locating information.

To help nodes form small-words, we group nodes by their ontological interest. In an open and dynamic global-scale grid it is infeasible to expect different users to conform to a global ontology; rather, nodes may have different ontologies. We propose an ontology directory service to organize these dispersed ontologies, and to allow nodes to efficiently locate their domains of interest. The directory facilitates the creation of ontological small worlds, and as a result, the queries can be forwarded to nodes that are likely to contain the relevant resource. This organization leads to a lower network load and a better search performance.

3.2. The Two-Tier Structure

3.2.1. Directory Overlay

As shown in Figure 1, the network is maintained and organized by employing two layers – the directory layer and the community layer; and the former facilitates the formation of the latter. We use a P2P-based DHT overlay to implement the directory service. The DHT overlay enables the communities to be built efficiently without any central control and creates meeting points for nodes with similar semantics. Based on the semantic interests, a new node will register its interest to the directory overlay. The overlay node in charge of the interest will provide a list of contacts, such that new node can join the community by connecting to those contacts. A node with multiple interests can register with multiple communities.

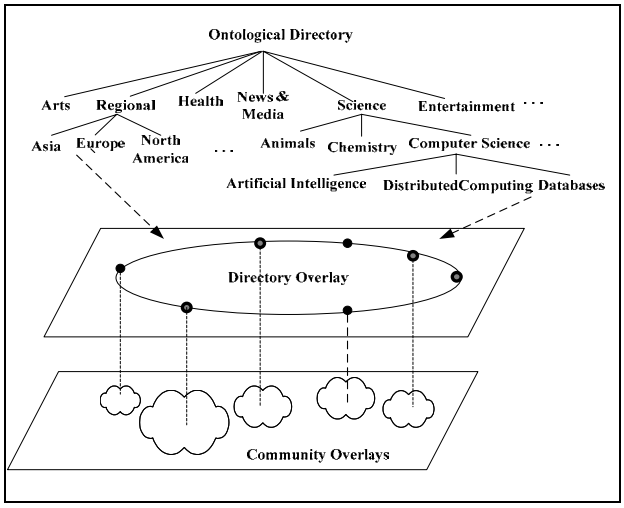

Figure 1. The two-tier structure.

The system provides users two registration interfaces: registering by keyword lookup or by browsing the directory. A user can specify one or more key concepts in its ontology and use the concepts as keys to lookup an existing ontology directory entry. The node that is in charge of this keyword keeps links to the corresponding directory entry. If multiple entries appear under the same keyword, these are returned to the user for further specification. Another way of registering is through the online directory browser. Users can dynamically retrieve the directory entries they are interested in. To realize the browser, a DHT node that is in charge of a directory entry also stores its children information. When the user chooses one entry, DHT routes to that entry and retrieves child entry information, such that the directory can be extended dynamically while browsing.

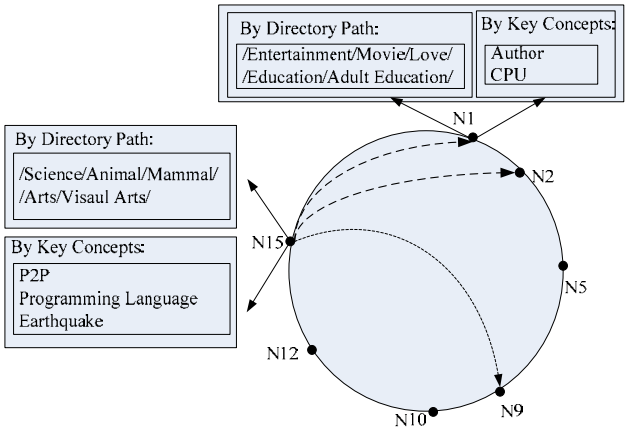

Figure 2 Directory Overlay Structure

3.2.2. Community Overlays

With the help of the directory overlay, nodes sharing similar ontological interests connect with each other and form community overlay networks. The topology of the community overlay can be flexible: if the community is large, nodes will be organized into a structured DHT overlay to efficiently distribute and locate resource knowledge; if the community is small, to avoid the cost of maintaining a DHT, nodes form an unstructured Gnutella-like overlay. The next section explains to efficiently manage and discover resource knowledge in a relatively large community using the DHT approach. Integration in a small community can be achieved using a similar gossip-based strategy applied to a Gnutella-like overlay.

4. Knowledge Management

4.1. Knowledge Repository

Inside the community, each node uses an ontology to explicitly describe details of each possessed resource, as well as its knowledge of concepts and relationships regarding its resources. The knowledge of an ontology is separated in two parts: the terminological box (T-Box) and the assertion box (A-Box). The T-Box statements describe concepts and relationships between concepts, for example a *Computer contains Memory*. The A-Box represents the concrete knowledge about individual instances of concepts, and their relations to other individuals, for example that *PC103 contains 132MB RAM*.

GODIS enables shared access to an aggregated community ontology, where each organizational node administers its local knowledge concepts and data.

4.2. Ontology Integration

Currently, organizations construct ontologies separately, often for the same field of knowledge; thus, we expect nodes in our communities to independently develop local ontologies. When joining a community, a node has to integrate its ontology's conceptual data (T-box) with the community knowledge. The instance data remains localized, and is integrated dynamically by the query engine as will be shown in Section 4.4.

4.2.1. Ontology Differences

When different ontologies are to be integrated, several issues must be considered such as incompatibility between underlying representation languages, data encoding mismatches, differences in modeling events, concept granularity and specialization, and concept terminology differences as affected by synonyms and homonyms.

In the GODIS system, it is assumed that all ontologies adhere to the OWL Lite standard [18]. While the representational mismatch issues may be solved by trivial transformations, the other outlined issues may require manual intervention. Several popular semi-automatic tools such as Protégé-PROMPT [21] and Chimaera [22] are available for eased ontology integration. Prior to officially joining the community, users can make use of these tools and bring their ontology in partial agreement with the community. When a node enters the community, its knowledge concepts, relations and instances remain intact, and appropriate mappings are made to the community ontology.

4.2.2. Inter-Ontology Relationships

The system allows OWL Lite relations between different ontologies. For example, a node may create relations between concepts to denote *equivalence* and *subclass*; and between properties to denote *equivalence* and *inverse*. We note that various nodes may contain different supplementary information about the same real world individual; thus we add a special *supplements* relation between concepts. This allows individuals to be merged if specific properties match, creating an aggregated entity. The logical relations of *equivalence* and *subclass* are extended to operate on such aggregated individuals.

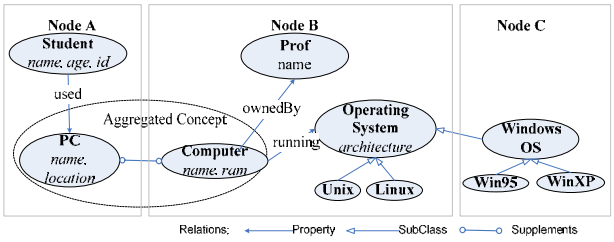

Figure 3. Integrated Ontologies

Figure 3 shows the resulting terminological space when three nodes are integrated. Node B and Node C relate through a *subclass* relation, allowing Node C to match individuals when the "OperatingSystem" concept is queried. A *supplements* relation is also created between the Node A "PC" concept and Node B "Computer" concept. This states that the classes describe different aspects of the same higher level concept, and their instances may overlap. They are termed *supplementary concepts* based on the property "name"; thus instances of

these different classes can be aggregated if they have the same "name". Figure 4 shows an individual instance of the aggregated concepts.

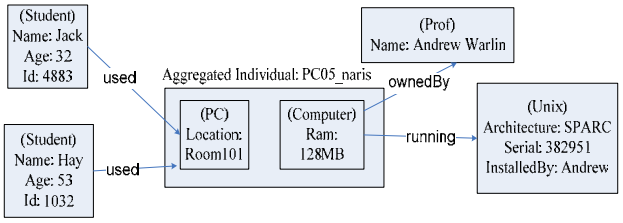

Figure 4. Aggregated Individual

The concept and property inter-nodal relationships are stored in the community overlay, allowing efficient processing in the inference and query engines.

4.2.3. Community Knowledge Overlay

A DHT overlay structure is used to track the inter-ontology assertions outlined above. We store each knowledge assertion twice, indexing by the subject and object of an assertion triple. Each node sends out its own assertion to the responsible nodes in the DHT. The target DHT nodes store the assertion and possibly generate new assertions through their inference engine, and the new assertions will be inserted into the DHT as well. Thus, after finishing this process, the knowledge is accessible in a well-defined way over the DHT network.

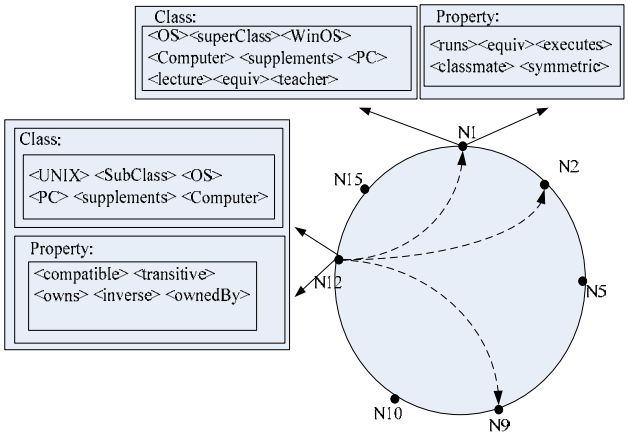

Figure 5 Community Knowledge Overlay Structure

It is also possible to store the instances into the overlay, such that instances can be located quickly at the cost of overlay storage space and data distribution.

4.3. Knowledge Expansion and Inference

A node can update its local knowledge ontology. Modifications which disturb relations between nodes are immediately updated in the community overlay, and affected nodes alerted through the overlay. To maintain administrative control over localized knowledge, a node cannot create relations between concepts outside its own local ontology except those mentioned in Section 4.2.2.

When an ontology is modified, it is possible for new relations to be inferred. At the conceptual level, transitive relations need to be extended to all affected classes. This kind of inference is performed once a node updates the knowledge base. Other kinds of inferred relations, such as transitive properties between individuals, will be dynamically determined during the query process.

4.4. Query Processing

Once a node has joined a community, the user can query information within that small world. A query can be created from scratch or with the aid of an ontology visualization tool.

4.4.1. Query Language

We use RDQL [23] as our query language, which is based on matching {subject, predicate, object} triples.

For example, the following query finds all operating systems whose architecture is SPARC.

```
SELECT ?x
WHERE ?x type OperatingSystem
      ?x architecture "SPARC"
```

The query can be efficiently distributed through the network when it is correlated with information from the community overlay. For the above example, only nodes B and C need to be investigated. Once an efficient query distribution path has been determined, the query propagates between network nodes, collecting and aggregating data.

4.4.2. Query Propagation

The individuals satisfying components of a query may be contained within only a cluster of nodes from the whole network. Thus the first step of query processing is to determine which nodes need to be traversed by the query, and in which order.

For example, the query "find all students that have used computers owned by Andrew" needs to traverse only nodes A and B.

```
SELECT ?student
WHERE ?student type Student
      ?student used ?comp
      ?comp type PC
      ?comp ownedBy ?prof
      ?prof name "Andrew Warlin"
```

A query will return one or more individuals which match specific constraints. An individual belongs to the result set when there is an agreement between query constraints and (1) the values of the individual's properties (eg: *'?prof name "Andrew Warlin"'*), and (2) relations between the individual and other individuals (eg: *'?student used ?comp'*). On a conceptual level, the nodal ontologies and relationships between them can be analyzed to determine which nodes may contain

individuals that satisfy the queried properties and relationships.

If a query extends between nodes, it must be separated into pieces that are executable within individual nodes, such that the query is partially processed within a local node and results forwarded to nodes capable of processing the rest, similar to a relational database "join" operation. In the above example, the *ownedBy* relation is known by node A, while *used* is known node B. Therefore the component, '*?student used ?comp*', must be executed on node A and the individuals matching the $comp variable must be forwarded to node B for processing of the '*?comp ownedBy ?prof*' component.

The order of steps in which a query is executed can greatly affect the performance of a search. Approaches to query optimization have been thoroughly researched [19, 20]. The query analysis engine must use the ontological knowledge to efficiently navigate the search space as to reduce the data that is aggregated at each inter-nodal junction in query processing. In the above example, processing '*?prof name "Andrew Warlin"*' before '*?comp ownedBy ?prof*' will drastically reduce the search space and amount of data transferred between nodes.

Once a query itinerary has been determined, the query can be executed by sending to the first node in its path. However, to insert the query into a node, the query has to be transformed to satisfy the local vocabulary.

4.4.3. Inter-Node Transformations

Queries may extend between nodes whose ontologies are related. Using the community overlay, a sending node can transform the query such that less processing is done at the receiving end. Specifically, the query can be translated to refer to the destination names for an equivalent concept or property, an explicit name for a subclass or superclass concept, or the query component may be inversed and renamed to match inverse properties. For example, prior to sending the above query to node B, the concept name *PC* is renamed to *Computer* to match the local ontology.

4.4.4. Data Extraction

Once the query is inserted into the network, data extraction and aggregation occurs. As a query propagates, partial results are acquired from nodes. The node processes parts of the query constraints individually as specified by the itinerary, and data matched may be part of the final result or simply an intermediate used to match relations in other nodes.

Since a node determines its individuals' agreement with the query property constraints, only individual aggregation identifiers need to be transferred between nodes for the process of joining results.

Once the unique individuals satisfying a query have been determined, only identifiers are returned to the user. To gather the full data about an entity, information must be aggregated from all nodes holding information about a specific individual and this is once again realized by navigating the concept space of the overlay.

5. Conclusion

As more and more resources appear in grids, there is a compelling need to find an effective and efficient way to discover and query these resources. In this paper, we presented GODIS, an ontological framework for resource integration and discovery in large-scale grids. The system provides a distributed ontological directory service to help nodes form communities according to their semantic properties. As a result, searching cost is reduced by sending discovery requests only to appropriate semantic clusters. Inside a community, nodes may use different ontologies to represent their resource and knowledge, and a distributed integration mechanism was proposed to cope with the mediation between different ontologies. We are currently prototyping an ontology-based grid resource search engine based on the techniques proposed.

References

[1] Watts, D.J., Strogztz,S,: Collective dynamics of 'small-world' networks. Nature 393 (1998).

[2] Globus Toolkit: http://www.globus.org/toolkit/

[3] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. "Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing," Technical Report, UCB/CSD-01-1141, April 2000.

[4] A. Rowstron and P. Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, Middleware, November 2001.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H.Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," ACM SIGCOMM, August 2001, pp. 149-160.

[6] S. Ratnasamy, P.Francis, M.Handley, R.Karp, and S. Shenker. "A Scalable Content-Addressable Network," ACM SIGCOMM, August 2001, pp. 161-172.

[7] Berners-Lee T, Hendler J, Lassila O (2001) The semantic web . Sci Am 284(5):34–43

[8] Stevens,R.D., Robinson,A.J. and Goble,C.A. (2003) myGrid: personalized bioinformatics on the information grid. Bioinformatics, 19, i302–i304.

[9] OntoGrid project: http://www.ontogrid.net/

[10] M. Arumugam, A. Sheth, and I. B. Arpinar. Towards peer-to-peer semantic web: A distribuited environment for sharing semantic knowledge on the web. In Proc. of the International World Wide Web Conference 2002 (WWW2002), Honolulu, Hawaii, USA, 2002.

[11] S. Castano, A. Ferrara, S. Montanelli, and D. Zucchelli. Helios: a general framework for ontology-based knowledge sharing and evolution in P2P systems. In IEEE Proc. of DEXA WEBS 2003 Workshop, Prague, Czech Republic, September 2003.

[12] Ora Lassila and Ralph R. Swick, "W3C Resource Description framework (RDF) Model and Syntax Specification".

[13] Dan Brickley and R.V.Guha. "W3C Resource Description Framework (RDF) Schema Specification".

[14] M. Cai, M. Frank, J. Chen and P. Szekely, "MAAN: A Multi-Attribute Addressable Network for Grid Information Services." The 4th International Workshop on Grid Computing, 2003.

[15] Iamnitchi A, Foster I, "On Fully Decentralized Resource Discovery in Grid Environments," Proc. The 2nd IEEE/ACM International Workshop on Grid Computing 2001, Denver, November 2001.

[16] M. Ehrig, C. Tempich, J. Broekstra, F. van Harmelen, M. Sabou, R. Siebes, S. Staab, and H. Stuckenschmidt. SWAP - Ontology-based Knowledge Management with Peer-to-Peer Technology. In Proceedings of the 1st National Workshop Ontologie-basiertes Wissensmanagemen, 2003.

[17] I. Horrocks, F. van Harmelen, and P. Patel-Schneider. DAML+OIL. http://www.daml.org/2001/03/daml+oil-index.html, March 2001.

[18] W3C. Web-ontology (webont) working group. http://www.w3.org/2001/sw/WebOnt/.

[19] Matthias Jarke, Jürgen Koch: Query Optimization in Database Systems. ACM Comput. Surv. 16(2): 111-152(1984)

[20] KING, J 1981. QUIST: A system for semantic query optimization in relational databases In Proceedings 7th International Conference on Very Large Data Bases (Cannes, France). VLDB Endowment, 510–517.

[21] N.F. Noy and M.A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, 2000.

[22] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000). Morgan Kaufmann Publishers, San Francisco, CA, 2000.

[23] A. Seaborne. "RDQL: A Data Oriented Query Language for RDF Models." http://www-uk.hpl.hp.com/people/afs/RDQL/