Towards a Scalable Grid Ontology Repository

Marian Babik and Ladislav Hluchy
Institute of Informatics, Dept.of Parallel and Distributed Computing, Bratislava, Slovakia
Marian.Babik@saske.sk, hluchy.ui@savba.sk

Abstrakt

This paper describes a scalable grid ontology repository system (OntStore), which unlike other ontology repositories is capable of storing and querying the ontologies and the resource descriptions in a distributed manner. It is based on the distributed hash table (DHT) and well known W3C standardized languages OWL/RDF/RDFS. The sys- tem decomposes the ontologies into the corresponding triples (subject, predicate, object) and uses the DHT system called Pastry to hash and store the elements. Results from experiments describing semantic data distribution and query routing performance will be presented. We'll also address the decomposition and indexing of the ontologies.

1. Introduction

Sharing semantically rich and heterogeneous data is one of the today's major challenges. Publishing and authoring web pages is quite easy, also sharing through the peer-to-peer (P2P) systems makes dissemination of the files a simple job. However there is no similar technology for sharing information with different schemas and providing efficient search and inference capabilities in a distributed manner. Yet, benefits of the semantic data sharing is enormous. Considering scientific data, until recently researchers analyzed and collected data in isolation. With emerging grid technologies and globally scalable systems, integrating and aggregating data from multiple sources and providing efficient search mechanism is necessary. Example of such global scale scientific pro jects include LHC [1], Human Brain Project, Institute for Systems Biology [2] etc.

In the recent grid pro jects using kwowledge based systems in composition and execution of complex worklows is becoming very popular. Among the advantages of such approach are more flexibile collaborative environments, possibility to compose dynamic workflows, which are faster, reliable and can cover broader range of application domain as well as possibility to learn from past experiences. Such system can enhance the possibilities of both application domain services and grid middleware services. However for such systems to work, it is necessary to share semantically rich data represented by a markup language. Such language has to describe an upper ontology for grid services and the corresponding groundings. Since the amount of the information, which has to be shared in grids is enormous, finding distributed ways of sharing the data is indeed necessary.

2. Design

OntStore is a system for scalable distributed storage of the ontologies represented in the OWL. Considering the fact, that centralized ontology repositories have limitations in their failure tolerance and their scalability, search for the possible distributed alternatives is indeed inevitable. Furthermore the grid applications are promising to host vast number of the web services providing an excellent environment to study and use the distributed storage technologies.

OWL is a well known language for defining and instantiating ontologies. OWL is based on well known Resource Description Framework (RDF), its schema extension RDFS and XML Schema datatypes (XSD). By providing additional vocabulary along with formal semantics, OWL facilitates greater interpretability than RDF/RDFS [4].

OntStore provides self-organizing and content addressable network based on the DHT systems. DHT supports location of resources based on their unique key. Our distributed network is based on the DHT system called Pastry [9]. Pastry functions as an overlay on top of the internet protocol (IP), using distributed, fault-tolerant data structure to explicitly track the location of all the resources. This structure extends the hashed-suffix routing structure presented by Plaxton, Rajaraman and Richa [7].

Pastry is based on the assumption, that nodes and messages can be identified with unique identifiers, represented as strings of digits (e.g. SHA-1 hashing). Identifiers are uniformly