Search Engines for the Grid: A Research Agenda

Marios Dikaiakos[1], Yannis Ioannidis[2], and Rizos Sakellariou[3]

[1] Department of Computer Science, University of Cyprus, Nicosia, Cyprus
mdd@ucy.ac.cy
[2] Department of Informatics and Telecommunications, Univ. of Athens, Greece
yannis@di.uoa.gr
[3] Department of Computer Science, University of Manchester, Manchester, UK
rizos@cs.man.ac.uk

Abstract. A preliminary study of the issues surrounding a seach engine for Grid environments, GRISEN, that would enable the provision of a variety of Grid information services, such as locating useful resources, learning about their capabilities, expected conditions of use and so on. GRISEN sits on the top of and interoperates with different underlying Grid middleware and their resource discovery mechanisms. The paper highlights the main requirements for the design of GRISEN and the research issues that need to be addressed, presenting a preliminary design.

1 Introduction

The Grid is emerging as a wide-scale, distributed computing infrastructure that promises to support resource sharing and coordinated problem solving in dynamic, multi-institutional Virtual Organisations [10]. In this dynamic and geographically dispersed setting, *Information Services* are regarded as a vital component of the Grid infrastructure [5,14]. Information Services address the challenging problems of the discovery and ongoing monitoring of the existence and characteristics of resources, services, computations and other entities of value to the Grid. Ongoing research and developments efforts within the Grid community are considering protocols, models and API's to provide an information services infrastructure that would allow efficient resource discovery and provision of information about them [5,12,6,14].

However, the identification of interesting and useful (in the user's context) resources can be a difficult task in the presence of too many, frequently changing, highly heterogeneous, distributed and geographically spread resources. The provision of information-services components, as currently envisaged by the Grid community [5], is a first step towards the efficient use of distributed resources. Nevertheless, the scale of the envisaged Grids, with thousands (or millions) of nodes, would also require well defined rules to classify the degree of relevance and interest of a given resource to a particular user. If one draws on the experience from the World Wide Web (arguably, the world's largest federated information system), efficient searching for information and services in such an environment will have to be based on advanced, sophisticated technologies that are automatic,

F. Fernández Rivera et al. (Eds.): Across Grids 2003, LNCS 2970, pp. 49–58, 2004.

continuous, can cope with dynamic changes, and embody a notion of relevance to a user's request. In the context of the WWW, this role is fulfilled by search engines [2].

The vision of this paper is that the technology developed as part of web search engine research, along with appropriate enhancements to cope with the increased complexity of the Grid, could be used to provide a powerful tool to Grid users in discovering the most relevant resources to requests that they formulate. Thus, our primary objective is to study issues pertaining to the development of search engines for the Grid. An additional objective is to design a search engine for Grid environments, named GRISEN, which can facilitate the provision of a wide range of information services to its users and can make this transparent from the particular characteristics of the underlying middleware. GRISEN is not intended to act as a substitute of existing systems for resource discovery, resource management or job submission on the Grid. Instead, GRISEN is expected to be a high-level entry point for the user for locating useful resources, learning about their capabilities, expected conditions of use, and so on, providing a unified view of resource information regardless of any possible different middlewares. This way, users can pinpoint an appropriate set of Grid resources that can be employed to achieve their goals, before proceeding with firing their application.

The remainder of the paper is structured as follows. Section 2 states the problem that motivated this research. Section 3 sets the requirements of GRISEN. Section 4 presents the initial design for GRISEN's architecture. Section 5 highlights the issues that need to be addressed. Finally, Section 6 concludes the paper.

2 Background and Problem Statement

Grid environments were first developed to enable resource sharing between remote scientific organisations. As the concept evolved, information services have become an increasingly important component of software toolkits that support Grids.

A *Grid Information Service* is a software component of the Grid middleware that maintains information about *Grid entities*, i.e., hardware, software, networks, services, policies, virtual organizations and people participating in a Grid [5,9]. This information, which is encoded according to some data model, can be made available upon request by the Grid information service that provides also support for binding, discovery, lookup, and data protection.

From the outset, *Directories* have been adopted as a framework for deploying Grid Information Services. Typically, directories contain descriptive attribute-based information and are optimized for frequent, high-volume search and lookup (read) operations and infrequent writes [1]. Access to directories is provided via Directory Services, which wrap directory-based repositories with protocols for network access and mechanisms for replication and data distribution. Globus information services, for instance, are provided by the Metacomputing Directory Service (MDS) [9,5], which is based on the Lightweight Directory Access Protocol

(LDAP) [20,17,15]. The goal of MDS is to allow users to query for resources by name and/or by attributes, such as type, availability or load. Such queries could be of the sort of "Find a set of Grid nodes that have a total memory of at least 1TB and are interconnected by networks providing a bandwidth of at least 1MB/sec" or "Find a set of nodes that provide access to a given software package, have a certain computational capacity, and cost no more than x," and so on. Along similar lines, the Unicore Grid middleware [7] publishes static information about resources. Users annotate their jobs with resource requirements; a resource broker, currently being developed for the EC-funded EuroGrid project will match user-specified requirements with available resources.

However, the means used for publishing resource information, in either Globus or Unicore, do not aim to support sophisticated, user-customized queries or allow the user to decide from a number of different options. Instead, they are rather tied to the job submission needs within the particular environment. As we move towards a fully deployed Grid — with a massive and ever-expanding base of computing and storage nodes, network resources, and a huge corpus of available programs, services, and data — providing an effective service related to the availability of resources can be expected to be a challenging task. If we draw from the WWW experience, the identification of interesting resources has proven to be very hard in the presence of too many dynamically changing resources without well-defined rules for classifying the degree of relevance and interest of a given resource for a particular user. Searching for information and services on the Web typically involves navigation from already known resources, browsing through Web directories that classify a part of the Web (like Yahoo), or submitting a query to search engines [2].

In the context of the Grid, one can easily envisage scenarios where users may have to 'shop around' for solutions that satisfy their requirements best, use simultaneously different middlewares (which employ different ways to publish resource information), or consider additional information (such as, historical or statistical information) in choosing an option. The vision of this paper is that search engine technology, as has been developed for the WWW, can be used as a starting point to create a high-level interface that would add value to the capabilities provided by the underlying middleware.

3 Requirements

A search engine for resource discovery on the Grid would need to address issues more complex and challenging than those dealt with on the Web. These issues are further elaborated below.

Resource Naming and Representation. The majority of searchable resources on the World-Wide Web are text-based entities (Web pages) encoded in HTML format. These entities can be identified and addressed under a common, universal naming scheme (URI). In contrast, there is a wide diversity of searcheable "entities" on the Grid with different functionalities, roles, semantics,

representations: hardware resources, sensors, network links, services, data repositories, software components, patterns of software composition, descriptions of programs, best practices of problem solving, people, historical data of resource usage, virtual organizations. Currently, there is no common, universal naming scheme for Grid entities.

In MDS, Grid entities are represented as instances of "object classes" following the hierarchical information schemas defined by the Grid Object Specification Language (GOS) in line with LDAP information schemas [18,15]. Each MDS object class is assigned an *optional* object identifier (OID) that complies to specifications of the Internet Assigned Numbers Authority, a description clause, and a list of attributes [16,18]. The MDS data model, however, is not powerful enough to express the different kinds of information and metadata produced by a running Grid environment, the semantic relationships between various entities of the Grid, the dynamics of Virtual Organizations, etc. Therefore, relational schemas, XML and RDF are investigated as alternative approaches for the representation of Grid entities [6,19,11]. Moreover, the use of a universal naming scheme, along with appropriate mapping mechanisms to interpret the resource description convention used by different middlewares, would allow a search engine for the Grid to provide high-level information services regarding resources of different independent Grids that may be based on different middlewares.

Resource Discovery and Retrieval. Web search engines rely on Web crawlers for the retrieval of resources from the World-Wide Web. Collected resources are stored in repositories and processed to extract indices used for answering user queries [2]. Typically, crawlers start from a carefully selected set of Web pages (a seed list) and try to "visit" the largest possible subset of the World-Wide Web in a given time-frame crossing administrative domains, retrieving and indexing interesting/useful resources [2,21]. To this end, they traverse the directed graph of the World-Wide Web following edges of the graph, which correspond to hyperlinks that connect together its nodes, i.e., the Web pages. During such a traversal (crawl), a crawler employs the HTTP protocol to discover and retrieve Web resources and rudimentary metadata from Web-server hosts. Additionally, crawlers use the Domain Name Service (DNS) for domain-name resolution.

The situation is fundamentally different on the Grid: Grid entities are very diverse and can be accessed through different service protocols. Therefore, a Grid crawler following the analogy of its Web counterpart should be able to discover and lookup all Grid entities, "speaking" the corresponding protocols and transforming collected information under a common schema amenable to indexing. Clearly, an implementation of such an approach faces many complexities due to the large heterogeneity of Grid entities, the existence of many Grid platforms adopting different protocols, etc.

Globus seeks to address this complexity with its Metacomputing Directory Service [5]. Under the MDS approach, information about resources on the Grid is extracted by "information providers," i.e., software programs that collect and organize information from individual Grid entities. Information providers extract information either by executing local operations or contacting third-party

information sources such as, the Network Weather Service or SNMP. Extracted information is organized according to the LDAP data model in LDIF format and uploaded into LDAP-based servers of the Grid Resource Information Service (GRIS) [16,17]. GRIS is a configurable framework provided by Globus for deploying core information providers and integrating new ones.

GRIS servers support the Grid Information Protocol (GRIP), an LDAP-based protocol for discovery, enquiry and communication [5]. GRIP specifies the exchange of queries and replies between GRIS servers and information consumers. It supports discovery of resources based on queries and information retrieval based on direct lookup of entity names. GRIS servers can register themselves to aggregate directories, the Grid Index Information Services (GIIS). To this end, they use a soft-state registration protocol called Grid Registration Protocol (GRRP). A GIIS can reply to queries issued in GRIP. Moreover, a GIIS can register with other GIIS's, thus creating a hierarchy of aggregate directory servers. End-users can address queries to GIIS's using the GRIP protocol.

Nevertheless, MDS does not specify how entities are associated with information providers and directories, what kinds of information must be extracted from complex entities, and how different directories can be combined into complex hierarchies. Another important issue is whether information regarding Grid entities that is stored in MDS directories is amenable to effective indexing. Finally, as the Grid scales to a large federation of numerous, dispersed resources, resource discovery and classification become a challenging problem [12]. In contrast to the Web, there is no global, distributed and simple view of the Grid's structure that could be employed to drive resource discovery and optimize replies to user queries.

Definition and Management of Relationships. Web-page links represent implicit semantic relationships between interlinked Web pages. Search engines employ these relationships to improve the accuracy and relevance of their replies, especially when keyword-based searching produces very large numbers of "relevant" Web pages. To this end, search engines maintain large indices capturing the graph structure of the Web and use them to mine semantic relationships between Web resources, drive large crawls, rate retrieved resources, etc. [4,2].

The nature of relationships between Grid entities and the representation thereof, are issues that have not been addressed in depth in the Grid literature. Organizing information about Grid resources information in hierarchical directories like MDS implies the existence of parent-child relationships. Limited extensions to these relationships are provided with cross-hierarchy links (references). However, traversing those links during query execution or indexing can be costly [14]. Alternatively, relationships can be represented through the relational models proposed to describe Grid monitoring data [6,8].

These approaches, however, do not provide the necessary generality, scalability and extensibility required in the context of a Grid search engine coping with user-queries upon a Grid-space with millions of diverse entities. For instance, a directory is not an ideal structure for capturing and representing the transient and dynamic relationships that arise in the Grid context. Furthermore, an MDS

directory does not capture the composition patterns of software components employed in emerging Grid applications or the dependencies between software components and data-sets [3,13]. In such cases, a Search Engine must be able to "mine" interesting relationships from monitoring data and/or metadata stored in the Grid middleware. Given that a Grid search engine is expected to be used primarily to provide summary information and hints, it should also have additional support for collecting and mining historical data, identifying patterns of use, persistent relationships, etc.

The Complexity of Queries and Query Results. The basic paradigm supported by Search Engines to locate WWW resources is based on traditional information retrieval mechanisms, i.e., keyword-based search and simple boolean expressions. This functionality is supported by indices and dictionaries created and maintained at the back-end of a search engine with the help of information retrieval techniques. Querying for Grid resources must be more powerful and flexible. To this end, we need more expressive query languages, that support compositional queries over extensible schemas [6]. Moreover, we need to employ techniques combining information-retrieval and data-mining algorithms to build proper indexes that will enable the extrapolation of semantic relationships between resources and the effective execution of user queries.

Given that the expected difficulty of queries ranges from that of very small enquiries to requests requiring complicated joins, intelligent-agent interfaces are required to help users formulate queries and the search engine to compute efficiently those queries. Of equal importance is the presentation of query results within a representative conceptual context of the Grid, so that users can navigate within the complex space of query results via simple interfaces and mechanisms of low cognitive load.

4 GRISEN Architecture

The effort needed to provide adequate information services on the Grid can partly be leveraged by considering, as a starting point, existing search engine technologies, which are subsequently enhanced with appropriate models and mechanisms to address the problems discussed above. In our envisaged Grid Search Engine, GRISEN, crawlers crawl the Grid collecting meta-information for Grid resources and policies thereof. Collected information is organized by a number of constructed indexes representing semantic and policy information about each resource. Access to GRISEN is provided to users through an intelligent-agent interface enabling simple, keyword-based searches and more complicated queries that take as arguments user-needs and preferences.

The purpose of GRISEN is not to change the existing layered architecture of the Grid or to substitute systems at each layer. Instead, GRISEN provides a universal interface that sits on the top, exploits the information provided by the layers underneath, and can be used by users to pinpoint a set of Grid resources

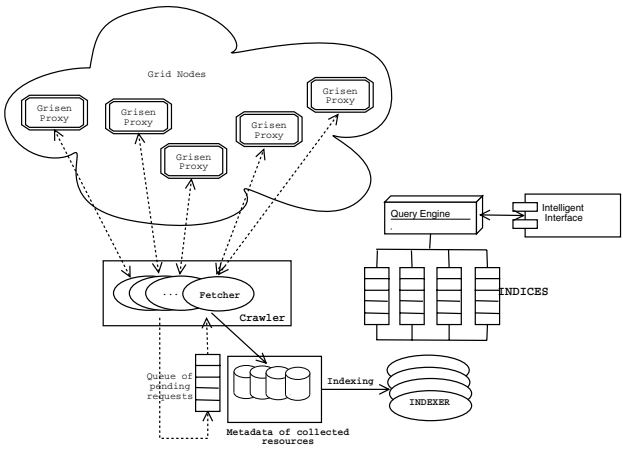

Fig. 1. Architecture of GRISEN.

that can be employed to achieve their goals, before proceeding with firing their application and invoking the necessary tools and services at the collective layer.

The architecture of GRISEN is established upon the notion of a Grid resource, which is represented by a complex data model, defined in the context of GRISEN. For the metadata of a Grid node to be retrievable by GRISEN, it has to become available at an information provider node or proxy, wherefrom it will be fetched by GRISEN.

Therefore, each Grid node is expected to have a corresponding proxy. Typically, the proxy is closely integrated with the node, in order to facilitate the efficient generation and publishing of the node's metadata. The proxy achieves this with the invocation of enquiry mechanisms provided at the fabric layer of the Grid. Some proxies, however, provide information about a wider range of resources belonging to different nodes under, for instance, the information directory service of a common administrative domain. Alternatively, proxies provide information regarding resources that span throughout different administrative domains but are perceived as a single subsystem or family of resources by application developers and users (e.g., a network connection, an index of CFD software, etc). Finally, a set of Grid resources can be represented by more than one proxy, each of which may provide complementary information.

Consequently, and due to the variety of existing and emerging Grid platforms, protocols and middleware, the specification of the proxy must comply with only a minimum set of requirements that enable the retrieval of metadata about the corresponding Grid resource from its proxy, via well defined, open protocols. Furthermore, each proxy should be uniquely identifiable via a universal naming scheme, possibly compliant to the naming schemes of the WWW (URL).

GRISEN consists of five basic modules: (i) **Proxies** distributed throughout the Grid, running query mechanisms at the fabric layer of the Grid to extract information about local resources. (ii) The multi-threaded, distributed "**crawler**"

that discovers and accesses proxies to retrieve metadata for the underlying Grid resources, and transform them into the GRISEN data-model. (iii) The **indexer**, which processes collected metadata, using information retrieval and data mining techniques, to create indexes that can be used for resolving user queries. (iv) The **query engine**, which recognizes the query language of GRISEN and processes queries coming from the user-interface of the search engine. (v) The **intelligent-agent interface** that helps users issue complicated queries when looking for combined resources requiring the joining of many relations. The overview of the whole system architecture is depicted in Figure 1.

5 The Context of GRISEN

GRISEN is expected to function in the context of a Grid viewed as a constellation of resources represented by "heterogeneous" information providers-proxies, with unique addresses, encapsulating meta-information about these resources in a common data model, and enabling the retrieval of this meta-information from remote hosts via well-defined protocols. To implement GRISEN in this context, the following issues need to be addressed:

Exporting Local Metadata into the Proxies: This issue refers to the extraction to the proxy of metadata describing the resources of a Grid node. Metadata must comply with a common data model to be defined in GRISEN. A specification of the proxy structure and interface to the Grid node are required; this interface must be compliant with the enquiry protocols implemented by various Grid platforms, different types of nodes, etc. A proxy can be many things: a simple index of local resources created by the Grid node, published on a Web server and accessed via HTTP; a daemon running on a network port of the node, awaiting for requests complying to a platform-specific protocol; a mobile agent launched by GRISEN and parked at a node hosting a directory of resources and running periodic queries to extract meta-information. Besides supporting the protocols of acknowledged Grid platforms like Globus or Unicore, GRISEN would need to employ a minimalist approach for defining, naming, addressing, and implementing proxies.

Discovery of Proxies: GRISEN must implement efficient mechanisms for discovery of Grid-resource descriptions throughout Internet. As a first step to facilitate the discovery process, GRISEN can explore naming schemes that can be adopted to identify proxies and Grid resources. Different approaches need to be studied and compared. For example: (i) periodic "crawling" of the Grid for the discovery of "passive" proxies, or (ii) updates of GRISEN structures by "active" proxies whenever metadata change. For the effective discovery of proxies in the presence of hundreds of thousands of Grid nodes on Internet, it is important to define and exploit the semantic and administrative "relationships" that will be established between Grid resources, as users exploit the benefits of the Grid and form dynamic Virtual Organizations, using multiple Grid nodes to solve a problem, coupling different codes, etc.

Retrieval of Metadata: Upon discovery of a proxy, GRISEN must retrieve and store its metadata for further processing. This simple task must be highly efficient and scalable with respect to the size of the Grid. Moreover, it is critical to incorporate proper scheduling, replacement and garbage-collection mechanisms in order to monitor and follow the rate of change of resources, to maintain the freshness of collected metadata, to achieve the prompt disposal of obsolete resources, etc.

Organization and Management of Data, Query Mechanisms and Interface: Collected metadata must be analyzed with techniques combining information-retrieval and data-mining algorithms to build proper indexes that will enable the extrapolation of semantic relationships between resources and the effective execution of user queries. A query language will be developed. Given that the expected difficulty of queries ranges from that of very small enquiries to requests requiring complicated joins, an intelligent-agent interface is required to help users formulate queries to the GRISEN data model.

6 Summary and Conclusion

The motivation for the ideas described in this paper stems from the need to provide effective information services to the users of the envisaged massive Grids. The main contributions of GRISEN, as it is envisaged, are expected to revolve around the following issues: a) The provision of a high-level, platform-independent, user-oriented tool that can be used to retrieve a variety of Grid resource-related information in a large Grid setting, which may consist of a number of platforms possibly using different middlewares. b) The standardization of different approaches to view resources in the Grid and their relationships, thereby enhancing the understanding of Grids. c) The development of appropriate data management techniques to cope with a large diversity of information

Acknowledgement

The first author wishes to acknowledge the partial support of this work by the European Union through the CrossGrid project (contract IST-2001-32243).

References

1. R. Aiken, M. Carey, B. Carpenter, I. Foster, C. Lynch, J. Mambreti, R. Moore, J. Strasnner, and B. Teitelbaum. Network Policy and Services: A Report of a Workshop on Middleware. Technical Report RFC 2768, IETF, 2000. http://www.ietf.org/rfc/rfc2768.txt.
2. A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. Searching the Web. *ACM Transactions on Internet Technology*, 1(1):2–43, 2001.
3. R. Bramley, K. Chiu, S. Diwan, D. Gannon, M. Govindaraju, N. Mukhi, B. Temko, and M. Yechuri. A Component based Services Architecture for Building Distributed Applications. In *Proceedings of the 9th IEEE International Symposium on High Performance Distributed Computing*, pages 51–59, 2000.

4. S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual (Web) Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
5. K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01)*, pages 181–194. IEEE Computer Society, 2001.
6. Peter Dinda and Beth Plale. A Unified Relational Approach to Grid Information Services. Global Grid Forum, GWD-GIS-012-1, February 2001.
7. D. W. Erwin and D. F. Snelling. UNICORE: A Grid Computing Environment. In *Lecture Notes in Computer Science*, volume 2150, pages 825–834. Springer, 2001.
8. Steve Fisher. Relational Model for Information and Monitoring. Technical Report, Global Grid Forum, 2001.
9. S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. In *Proceedings of the 6th IEEE Symp. on High-Performance Distributed Computing*, pages 365–375. IEEE Computer Society, 1997.
10. I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
11. D. Gunter and K. Jackson. The Applicability of RDF-Schema as a Syntax for Describing Grid Resource Metadata. Global Grid Forum, GWD-GIS-020-1, June 2001.
12. Adriana Iamnitchi and Ian Foster. On Fully Decentralized Resource Discovery in Grid Environments. *Lecture Notes in Computer Science*, 2242:51–62, 2001.
13. C. Lee, S. Matsuoka, D. Talia, A. Sussman, M. Mueller, G. Allen, and J. Saltz. A Grid Programming Primer. Global Grid Forum, Advanced Programming Models Working Group, GWD-I, August 2001.
14. B. Plale, P. Dinda, and G. von Laszewski. Key Concepts and Services of a Grid Information Service. In *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems (PDCS 2002)*, 2002.
15. Warren Smith and Dan Gunter. Simple LDAP Schemas for Grid Monitoring. Global Grid Forum, GWD-Perf-13-1, June 2001.
16. USC/ISI. *MDS2.2: User Guide*, 2 2003.
17. G. von Laszewski and I. Foster. Usage of LDAP in Globus. http://www.globus.org/mds/globus_in_ldap.html, 2002.
18. G. von Laszewski, M. Helm, M. Fitzerald, P. Vanderbilt, B. Didier, P. Lane, and M. Swany. GOSv3: A Data Definition Language for Grid Information Services. http://www.mcs.anl.gov/gridforum/gis/reports/gos-v3/gos-v3.2.html.
19. G. von Laszewski and P. Lane. MDSMLv1: An XML Binding to the Grid Object Specification. Global Grid Forum, GWD-GIS-002. http://www-unix.mcs.anl.gov/gridforum/gis/reports/mdsml-v1/html/.
20. W. Yeong, T. Howes, and S. Kille. Lightweight Directory Access Protocol. IETF, RFC 1777, 1995. http://www.ietf.org/rfc/rfc1777.txt.
21. D. Zeinalipour-Yazti and M. Dikaiakos. Design and Implementation of a Distributed Crawler and Filtering Processor. In A. Halevy and A. Gal, editors, *Proceedings of the Fifth Workshop on Next Generation Information Technologies and Systems (NGITS 2002)*, volume 2382 of *Lecture Notes in Computer Science*, pages 58–74. Springer, June 2002.