N. Nassar, Etymon
G. Newby, Arctic Region Supercomputing Center
K. Gamiel, CNIDR
M. Dovey, Oxford University e-Science Centre
J. Morris, CNIDR

August 2004

GGF Grid Information Retrieval Working Group
Category: INFORMATIONAL

Grid Information Retrieval Architecture

Status of this Memo

This memo provides information to the grid community regarding the architecture of Grid Information Retrieval (GIR) which is being developed by the Global Grid Forum GIR Working Group. Distribution of this memo is unlimited.

Copyright Notice

Abstract

Grid Information Retrieval (GIR) is a specification for general-purpose, interoperable, distributed information retrieval, using the computational grid as a common platform. The architecture described in this document seeks to address the requirements stated in GWD-R/draft-ggf-gir-requirements (the GIR "requirements document"), and proposes a model for distributed IR that draws from and builds on previous efforts toward IR interoperability. The architecture distributes the processes of creating and querying IR systems, and the systems themselves are federated and may be entirely decentralized. This paper outlines the architecture of GIR.

nassar@etymon.com

Contents

nassar@etymon.com

1. Introduction

During the last decade there has been a proliferation of information retrieval (IR) implementations and, correspondingly, IR architectures. The task of the GIR Working Group is to distill the most useful features of existing architectures into a single model that can serve as a common environment for interoperability of existing and future systems. The most difficult aspect of this, as often is the case in standards processes, is to determine the ideal convergence of ideas in order to avoid a specification that is either too limiting or too broad. The Working Group (WG) has sought input from many representative communities that implement IR systems, and it has attempted to balance this input with two primary areas of interest: IR research and practical IR interoperability. These two motivations are the "center of gravity" of the present work.

It should be noted at this point that our use of the term, "information retrieval," or "IR," is in the specific sense used by researchers and practitioners of information science, informatics, and computer science. Specifically, IR is concerned with automated methods of accepting queries from an information seeker and responding with documents that best satisfy the seeker's information need. This may include any variety of underlying data structures and data collections, although it is primarily intended for querying unstructured or semi-structured data (such as, for example, free text or XML-encoded text) as opposed to highly structured data (such as relational databases). GIR is not an IR implementation per se, but rather an architecture for communities of IR implementations to interoperate within.

We take as a starting point for GIR certain assumptions about the minimal requirements for a modern IR interoperability standard, and these are detailed in the GIR requirements document. There are many features or aspects of an IR system, some related to each other and some independent, and in general too many to approach in a completely organized manner. The WG has proposed that a GIR implementation be segmented into three fundamental IR tasks, or functional groups: (1) collecting documents into a local store, (2) indexing or other preparation of the documents, and (3) querying one or more document collections and associated retrieval of search results. The rest of this document assumes this particular model of a distributed IR system. Specifically, GIR defines interfaces that correspond roughly to each of the three fundamental IR tasks, respectively: (1) Collection Manager, (2) Index/Search, and (3) Query Processor.

GIR is proposed with the OGSA architecture in mind, and OGSA is intended as the target platform. At the same time we would like to describe the GIR architecture and specification in a way that is somewhat neutral in relation

nassar@etymon.com

to platform, by attempting to place OGSA-specific implementation issues in a separate section of this document.

2. Design Considerations

The principal design consideration of this particular approach to IR interoperability is a recognition of the "monolithic" character of most existing IR implementations and a desire to enable greater modularity in future IR systems. One of the primary motivations of GIR is a concern that IR systems are becoming too complicated, massive, and eclectic for monolithic implementation. In addition, there is an increasing need for both "on demand" and event-driven information sharing, and thus for IR systems to be able to work together intimately. Another reason for modularizing these systems is that one size does not fit all when it comes to IR. While a monolithic web search engine may be ideal for web searching, its limitations become evident when one would like to search data collections other than, or in addition to, the web. Finally, a successful federated IR implementation would significantly increase available processing power, since current monolithic systems are restricted by the need to provide instant response to queries of extremely large data collections.

It is also necessary to consider in the design of GIR the scalability of interactions over a network in much the same way that this would need to be considered in a more traditional distributed system.

An important design consideration is security throughout the system. It is an essential feature of GIR that each component within a virtual GIR organization be permitted a fine degree of resource control. This model combined with a distributed architecture can enable greater sharing of semi-public information among organizations by supporting their highly customized access constraints. This is one of the goals of GIR.

3. Architecture and Terminology

This section provides an overview of the GIR interfaces and the relationships among them. When we refer to a "GIR service," we mean a grid service that implements one or more GIR interfaces. In a simple case, each GIR service implements a single GIR interface, and our discussion generally assumes this simple case, so that we may talk about "a CM service" or "the QP," etc. There seems to be some confusion about how this is implemented in OGSA, but we expect that to be resolved over time.

The Collection Manager (CM) is concerned with collecting and managing source documents intended to be indexed, searched, and retrieved via one or more GIR services. It does this by retrieving specified documents from

various locations (remote or local), preprocessing and managing them in a local store, and providing them to clients according to specified rules. "Client" here normally refers to an Index/Search service, which uses CM's as the sources of documents that will be indexed and searched. Any CM can be requested to monitor its sources for updates and/or provide notification to Index/Search services when updates are available. In essence, the CM acts as a virtual document collection that feeds one or more Index/Search services. At the same time, an Index/Search service may retrieve documents from one or more CM's.

The Index/Search interface (IS) is therefore not troubled with the need to assemble documents from the variety of existing sources; the CM takes care of that. The documents are provided to the IS, and there the real work of indexing and/or otherwise preparing the documents begins. The IS creates and manages data structures needed to provide searching capabilities. (These data structures are normally referred to as an "index.") The IS also exposes a "search interface," which is a set of functions that allow searching of the indexed document collection. We choose to name this particular group of functions because the Query Processor also provides these functions, which means that queries through the search interface are supported by both the IS and the Query Processor. IS represents the core features of a traditional IR system. It accepts documents, indexes them, and provides searching capabilities on those documents. Furthermore it provides a common search interface, which is comparable to traditional IR standards such as Z39.50. The CM adds to this a distributed document collection and event-driven updates. The Query Processor distributes the searching side of the IR system.

The Query Processor (QP), the third architectural component of GIR, is responsible for managing queries and result sets (i.e. response sets from searching operations). It may preprocess a query before submitting it to an IS, for example, performing thesaurus expansion on the query. It may submit the query to more than one IS, in which case it is also concerned with merging result sets from those IS's.

As mentioned earlier, QP provides a search interface identical to that of IS. In other words, for purposes of searching, QP is a "virtual IS." It has no indexing capability of its own (and provides no indexing interface) but serves as a search-only gateway to one or more IS's. It is not our intention that a QP implementation have any searching capability of its own; however, since the search interface exists, it is possible to serve a locally indexed collection via QP. But it is preferable that such a system be served via an IS interface, even if no indexing functions are provided through GIR. Hopefully it is clear that a typical use of QP's would be to act as a primary interface to a collection of IS's and CM's that can be managed dynamically and independently of the QP's. The QP's may be the only access points, or

it may also be possible to access the individual IS's and CM's. The important concept is that any arrangement of CM's, IS's, and QP's is only one of potentially multiple virtual GIR organizations.

All GIR interfaces include a function called, "Explain." (The term is borrowed from Z39.50.) Explain provides metadata about a service, such as its supported capabilities, the type of content available, etc. For example, IS provides information about (1) what document collections are available for searching, (2) which subset of searching capabilities are supported, (3) what metadata elements are directly searchable, etc. All such information are accessed via the Explain function.

In general, the interactions among GIR services are asynchronous, and both the management/update and searching of a GIR service may be performed entirely through event-driven notification.

4. Components and Interfaces

This section provides further detail about the GIR interfaces.

Collection Manager

The CM may be considered as a black box with its main input and output consisting of documents. It brings documents into its local store and sends them off, usually to one or more IS components for indexing (but potentially to other CM components). Both the input and output interactions are controlled by "configuring" the CM with a set of rules.

This configuration is not necessarily done by the local system administrator, but by the authorized "administrator" of the GIR system, and it is performed via GIR function calls of the CM. It could also be done by a "client" component (and again, we mean an IS component); for example, an IS implementation could create a CM instance, or communicate with an existing one, and "configure" that CM itself. This is what we mean by configuration of GIR services, and for that reason it is not clear whether the traditional separation of "administrator" and "user" applies very well to a grid-based distributed system such as GIR. It is not only possible but desirable that "users" be able to create and augment IR systems within the GIR model, and the granularity of that control may vary depending on the virtual organization.

The "input" of documents into a CM is generally a process of harvesting those documents from a remote server, often via a TCP/IP protocol such as HTTP or FTP. However, the documents may also be sent to the CM as events, as in the case of a news feed, for example. The CM may also perform searches on QP or IS nodes and retrieve documents designated by

the result sets (or via the "standing query" function of QP). The CM is typically configured to harvest documents from specific servers (e.g. by URL), and optionally to check for updates on a regular schedule. The documents may be considered for retrieval according to certain criteria, with only conforming documents being selected; again this is specified as part of the configuration.

The "output" of documents from a CM to a client can be requested by the client or provided to the client as part of an event. This also is specified by configuration. If the CM is configured to "push" the documents to a client under certain circumstances, it will do so by sending notification to the client that new documents are available. "Under certain circumstances" means that the CM has received document updates, or the client has requested notification on a time schedule, or some other internal or external event has occurred that is understood by the CM implementation.

Index/Search

The IS component is of course concerned with indexing and searching. The implementations of these two functions are very closely coupled; however, they are architecturally rather distinct and easy to separate. The inclusion of an indexing function in GIR is fairly novel for an IR standard, and the approach we have taken to the indexing interface is based on the common implementation experience of the WG. The most significant feature of the indexing architecture is the distributing of the document collection to one or more CM components. Since some preprocessing or normalization can occur in the CM, the task of managing a variety of input documents and document sources in the IS may be reduced.

However, the IS interface is still the heaviest component of the GIR, with the most implementation complexity, the richest set of functions, and the largest number of combinations in which its functions may be used. For this reason, the most architecturally significant features of GIR are located in the CM and QP components, in order to distribute the overall complexity of implementation.

The IS indexing function is configured in the same way that CM is configured, with its relationship to CM being the only source of documents understood within the scope of GIR. IS may "poll" one or more CM's for documents and document updates, and/or it may receive notification of documents and document updates from one or more CM's. In either case, the IS can notify one or more specified QP's that the collection has changed, once its local index has been updated to reflect the changes. (That event might, for example, cause the QP to reissue a "standing query" and update its result set.) Other indexing features are specified in the configuration, such as re-indexing schedule, document types, indexing

options, and metadata elements to be indexed. The IS organizes indexed document collections into groups which we call “databases.” The databases may be searched individually or in combination. The list of databases available for searching as well as associated metadata are retrieved prior to searching via the Explain function.

The IS search interface is based on the Z39.50 IR standard (ANSI/NISO Z39.50, ISO 23950).

Query Processor

The QP is a point of entry for a client wishing to search a GIR system. It can represent a single IS or many IS components. The client can be a user client or another service, such as another QP.

As mentioned earlier, QP and IS share the same search interface. The main significance of this is that whenever we say that a QP queries or otherwise relates to an IS, the IS could in reality be either IS or another QP. QP is not concerned about whether it is a client of IS or QP, since both have an identical interface for purposes of searching. This allows distributed searching networks to be set up dynamically, by creating (normally) only a single QP. This is a flexible compromise between the traditional static model of distributed searching, in which one searches through a fixed entry point, and the Z39.50 model, which places all responsibility for the distributed search on the user client. Either approach could be implemented using the GIR interfaces, depending on the IR system requirements.

QP is dedicated to searching, and especially distributed searching. Its main purpose is to manage the processing and distribution of queries to multiple IS nodes and to reduce the responses into a single result set. It also issues distributed calls to the Explain function and merges those results to provide a single Explain response for its clients; which of course normally occurs before one begins issuing queries. Thus, from the point of view of a client interested in searching existing IR systems, QP is not very much different from IS. The real work occurs between QP and the IS/QP components to which it distributes queries and from which it merges results.

QP can optionally be configured to manage a “standing query,” by which we mean a query that is automatically reissued to an IS (on a schedule, or in the event of changes to the collection, or on the basis of some other criteria). If a standing query detects changes in the result set, it will notify the user client. This can also be used to implement “filtering,” an IR concept in which a fixed query is applied to a stream of documents and a Boolean relevance judgment is made for each document as being either

"relevant" or "not relevant."

We should note that documents themselves are not handled by QP or the search interface of IS. The search interface of both QP and IS return references to the documents, and in order to retrieve the documents, the client must either communicate with the CM that provided the documents or retrieve the documents directly via URL. The CM's and/or URL's needed for this purpose are included in the result sets.

5. Sample Use

This section describes a sample usage scenario of GIR, in order to explain how the architecture could be applied.

Gnewes Inc., a large news-gathering corporation, decides to expose a subset of its documents to external use via GIR. To avoid possible abuse, they require registration before allowing access to their CM. Other organizations, once they have registered, can connect their IS or QP components to the CM. Gnewes Inc. might also run an IS or QP.

The virtual organizations (VO's) consist of two members: one is Gnewes Inc. and the other is a registered organization. Larger VO's could be created if registered organizations wish to join together, but otherwise the VO's exist in pairs because there is no requirement by Gnewes Inc. that the registered organizations be part of the same VO. (For example, the registered organizations might be competitors.)

The registered organizations can then build up IS nodes or set up QP's and make their own decisions about who else can join in the VO's for those QP/IS groups. An end-user interested in searching the contents of Gnewes Inc. could set up his or her own QP that would connect to one of the registered organizations' IS nodes.

When a "hit" (i.e. document citation) is found by the user, he or she would select a document for viewing. The request would either go directly to Gnewes Inc. (if the user is part of Gnewes' VO) or to the registered organization's IS node, which would forward the request to Gnewes Inc. In both cases, Gnewes' CM would decide whether to provide the document, based on its access control list.

6. Security Considerations

Grid security issues relevant to this discussion are still evolving through existing GGF working groups. At the time of this writing, the GIR Working Group believes that the security implementation needed for GIR will be largely orthogonal to the GIR interface specification. GIR WG continues to

follow developments in the relevant security groups in the GGF.

7. OGSA Implementation

[To be written]

8. Author Information

Nassib Nassar
Etymon Systems, Inc.
P.O. Box 12484
Research Triangle Park, NC 27709-2484
USA
email: nassar@etymon.com

Gregory Newby
Arctic Region Supercomputing Center
P.O. Box 756020
Fairbanks, AK 99775
USA
email: newby@arsc.edu

Kevin Gamiel
Center for Networked Information Discovery and Retrieval (CNIDR)
3021 Cornwallis Road
Research Triangle Park, NC 27709-2889
USA
email: kgamiel@cnidr.org

Matthew J. Dovey
Oxford University e-Science Centre
OUCS
13 Banbury Road
Oxford OX2 6NN
England
email: matthew.dovey@oucs.ox.ac.uk

Jeremiah Morris
Center for Networked Information Discovery and Retrieval (CNIDR)
3021 Cornwallis Road
Research Triangle Park, NC 27709-2889
USA
email: jeremiah@cnidr.org

Acknowledgements

Many participants of the GIR Working Group have contributed significantly to GIR. The authors also wish to acknowledge the support of the National Science Foundation (Grant No. CCR-0082655). The views expressed in this paper are not necessary those of the funding organizations.

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

nassar@etymon.com

THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

nassar@etymon.com