

Grid Information Retrieval Requirements
June 5, 2003

Status of This Memo

This memo provides information to the Grid community regarding minimal and optional requirements for an OGSA grid-based information retrieval system. Distribution of this memo is unlimited.

Copyright Notice

Copyright © Global Grid Forum (2003). All Rights Reserved.

Abstract

GridIR is information retrieval on the OGSA grid. It is an effort to utilize grid computing advantages applied to the science of information retrieval in a unique manner. As with any new system design, it is important to understand stakeholder community requirements in order to develop a robust architecture and related specifications. This document details those requirements and establishes a base and context for future system architectures and specifications.

Table of Contents

1. Introduction.....	1
2. Notational Conventions	3
3. General Requirements.....	3
4. Security Considerations	6
5. Glossary	8
6. Author Information	8
7. Intellectual Property Statement.....	9
8. Full Copyright Notice	9
9. References.....	10

1. Introduction

GridIR is information retrieval (IR) in an OGSA grid-computing context [OGSA-PHY]. Though core grid-computing concepts and implementations continue to evolve, GridIR recognizes the inherent advantages provided by generalized grid computing, specifically as those advantages relate to the information retrieval problem space. GridIR expands on decades of experience with other distributed,

networked information retrieval systems such as Z39.50 [Z3950-95]. GridIR seeks to 1) define functionality and semantics required by a generalized information retrieval system, 2) describe an architecture based on a particular grid-computing platform, and 3) detail specifications suitable for building and operating a complete grid-based IR system. This document specifically addresses the requirements for a generalized IR system. It is expected that one or more architectures and associated specifications will result from this work.

GridIR is about moving forward to the days of network-aware search appliances, virtual organizations (VO) for information sharing, information security at multiple levels, and large-scale high-performance searching that is customized for particular collections and information needs. In short, it is about enabling today's sources for electronic information (such as Web servers, database servers, portals, etc.) to be integrated into highly secure, scalable information retrieval systems.

We are not skeptical of the future of large monolithic search engines, and hold them in high respect. However, we recognize the limitations of these systems, and wish to enable future information seekers to have choice among different information retrieval systems. We further wish to make the vast quantity of information that is currently inaccessible to search engines more readily available to information seekers.

The starting point for GridIR is the scientific discipline of information retrieval (IR). In IR, computer systems are used to match statements of human information need (a.k.a., queries) to documents. Statements of information need may be anything from a few words (typical of Web search engines) to a structured logic (e.g., Boolean statements) to a profile of the information seeker and her assessments of past documents.

In practice, IR seldom actually presents answers to questions, or even specific information items (such as a table of figures, or a quotation, or a particular passage in a document). Rather, IR systems present a ranked list of document citations in which, it is hoped, the desired information may be found. IR systems may be defined as those computer-based systems that take a number of documents (perhaps in different formats) as input and build data structures so that they may be quickly searched for matches to queries. These IR systems are able to then take queries (perhaps with special formatting or restrictions) as input and produce a list of documents in ranked order.

We call the set of documents input to an IR system a *collection*. For a given request, there might be more than one IR system that provides access to a collection of interest. While the days of massive search engines have led optimistic information seekers to think that all the information in the world is a click away via their favorite search engine, even these searchers know that different search engines host different collections. More experienced searchers know that there are thousands of different collections, different data types, different query languages, and so forth.

GridIR will help to make virtual organizations' collections more available to information seekers. Rather than seeking to harvest these collections into monolithic search engines, GridIR seeks to utilize distributed federated collections. There are important advantages over monolithic search engines in the GridIR scenario:

1. Queries will be run against collections of documents with a likelihood of possessing relevant documents, thus eliminating a priori those collections unlikely to have documents of interest. For example, someone interested in information about household pets might want to omit databases about taxidermy from consideration.
2. Each collection will be customized for that collection's qualities, according to the desires of the collection provider. Customization can include the full range of query processing, document processing and IR techniques such as document and term weights, IR retrieval models (Boolean, vector space, Latent Semantic Indexing, etc.), term stemming, stop word lists, etc.

3. Rather than seeking sub-second response times over billions of documents, as monolithic search engines do, GridIR systems may seek such performance over far smaller collections. This will enable more complex query processing.
4. GridIR collections, because of their smaller size and locality to a particular collection source (i.e., an organization's web server), can be updated more quickly, thus eliminating the delay among harvest runs exhibited by search engines. Collectively, we expect the capacity of GridIR to far exceed any search engine.
5. Grid computing offers a notification model in which events (such as the availability of new content) can trigger other events (such as evaluating a query against the new content). This model opens the door to standing queries, information filtering, and push (rather than pull) approaches to information dissemination.

GridIR will provide a framework whereby existing, traditional IR engines may be plugged in at various parts of the architecture in order to construct a new, higher-level IR system. That is, traditional IR systems do not typically perform as distributed engines (computationally or in terms of source data), but by plugging into the GridIR framework that engine (perhaps unknowingly) can become part of a larger, highly secure, computationally and data-distributed IR system.

Since grid computing typically includes the virtual organization (VO) concept at its core, GridIR inherits and supports fundamental VO characteristics, specifically the security framework. As addressed in individual requirements, GridIR must go a step beyond standard service-level security, however, into finely controlled application-specific security.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC-2119 [RFC 2119].

3. General Requirements

3.1 distributed

The system must be modular with clearly defined interfaces in support of leveraging traditional distributed computing advantages. The level of modularity depends on a particular software architecture, baseline network infrastructure, etc. For example, if one assumes a reliable, high bandwidth network, the system might support more, smaller modules that perhaps make the system more flexible, though more complex. This requirement presumes a robust communication method among the distributed system modules.

3.2 asynchronous notification

There must be a method of sending and receiving out-of-band messages among participating modules. For example, one module (the sink) might request notification from another module (the source) upon some action or event taking place. The sink might express a desire to know when the internal state of the source has changed by "subscribing" to the source. That expression of interest, or subscription, establishes a loose link between the two modules. When the internal state of the source module changes, it sends a

notification message to the sink.

3.3 event-driven operation

Modules must have the ability to immediately act upon asynchronous notification messages received from other modules.

3.4 Service persistence

Each module must allow persistence of operation. A service is considered persistent if it remains available for some period of time after initial creation in order to service subsequent operational requirements. Such service persistence implies methods for creation and explicit destruction of instances of a service by authorized parties. Typically, there will be other methods for destruction of service instances including inactivity timers, etc.

3.5 query persistence

Relevant modules may allow persistent queries. A persistent query is defined as a long-lived query against a particular set of indices and its result set may be updated perhaps via scheduling or triggered by an event such as index update. For example, if a new document becomes available to a collection, that may trigger dependent indexers to update themselves based on the updated collection. In turn, an updated index may notify dependent processes maintaining persistent queries, such that the final result may be updated dynamically.

3.6 metadata services

All modules must be introspective in terms of access mechanism as well as semantic description. For example, a module may provide a list of access methods and parameter data types via WSDL. It must also describe the contents of its document, collection, index, and query holdings both in a machine and human-usable manner, e.g. Dublin Core metadata records.

3.7 data collection description services

There must be methods to describe an abstract data collection to a management service. An abstract collection description may include specific URLs, spidering rules, transformation rules, etc. A data collection description is considered input to a process that results in a collection.

3.8 data collection scheduling services

There may be methods to schedule periodic collection updates, perhaps as an extension of the data collection description. For example, a data collection description may describe crawling a particular web site, an associated schedule might specify how often to re-crawl the site.

3.9 data collection delivery services

There must be a mechanism for reliable file transport of collected raw data from data collection modules. There may be methods of transporting partial sets of data, e.g. diffs.

3.10 document content type independent

The system must be capable of acting on arbitrary data formats (content types). Specifically, the system must be able to describe and transport arbitrary data, but individual modules do not have to support that

particular type of data. For example, a collection may consist of text/plain and text/html files, but an indexer may invoke standard diagnostic services to deny support of the text/html files. Typically, the required metadata services would provide a list of supported data formats.

3.11 document transformation capabilities

There may be services to transform documents from one content-type to another or to post-process documents based on a set of rules or parameters.

3.12 index generation services based on input collections

There must be a method of creating a searchable index for one or more data collections. How a collection is actually indexed, if at all, is outside the scope. In practice, there is a common denominator set of options and parameters for most indexer applications. This set should be reflected through the appropriate service interfaces and translated to a particular system as necessary.

3.13 index query services generating result sets

An index must have the ability to accept queries and respond with relevant result sets. A query is a representation of the user's information needs, typically expressed either as simple keywords or in a normalized, structured manner, for example common query language (CQL). A result set is a set of metadata describing documents matching the user query, suitable for subsequent document retrieval via related services.

3.14 query type independent

There must be a mechanism for specifying arbitrary query types. Practically speaking, a small number of query types, e.g. CQL, will actually be supported but there must be the ability to unambiguously describe arbitrary query types for special-purpose use and future expansion.

3.15 minimum query type

To assure interoperability, there must be at least one particular type of query supported by the system. The query type should leverage existing open standards. The selected query type must at least support keyword queries, arbitrarily complex Boolean queries, and attributes on individual elements of the query. Attributes must include at least a field identifier in support of structured queries.

3.16 result set delivery services

There must be a method of retrieving result sets from indexers. The service must allow a user to retrieve one or more ranges of records from the set. Each record returned is subject to the data presentation requirement listed below.

3.17 result set deletion

There must be a method of deleting a remote result set.

3.18 record presentation

For each record in a result set there should be a method of requesting parts of the record (e.g. Z39.50 element sets) in a particular format (e.g. mime content type). However, the record host need not honor the request.

3.19 index scan services

There may be indexer scanning services, a service for browsing the contents of an index, e.g. words contained in an index. A scanning service typically provides lists of words that actually appear in an index in support of better quality query formation.

3.20 peer-to-peer communication model

The data communication model must support peer-to-peer communications, for example to support arbitrary linkages among multiple participating (possibly persistent) modules in support of a given task.

3.21 client-server communication model

The traditional client-server communication model should be supported. For example, a simple client may only know how to send a single request to an index and receive a single response during a session.

3.22 distributed searching

The system must support the ability to distribute a single query across multiple search modules and expect standard behavior.

3.23 merging

The system must provide services to merge two or more result sets into a single result set. It is recommended that the merged result set remains orthogonal.

3.24 multi-lingual capable

The system must be capable of unambiguously describing the language of content data. However, a particular module does not need to support any particular language. For example, an indexer may know it has received German language content, but it may not know how to properly search such data and must issue relevant diagnostics.

4. Security Considerations

Information security is a major component for data sets. Security concerns might occur at the level of datum (i.e., a document), user or query source, or an entire set of data or its IR system. At the least, GridIR must provide a security level that assures the same level of security that is associated with the data in the system. With GridIR, issues of data channel and system-level security are largely met by the underlying OGSA infrastructure [OGSASec] and the ongoing work with Web Services Security [WSec]. Fundamentally, this means that a decision of whether to admit a particular system into your virtual organization is the first and coarsest security decision. A set of additional security challenges for GridIR is based on the security requirements of the information content.

Security management in information and network systems has three basic principles [SecurityHandbook] that are targeted as goals that a good security infrastructure should be able to fulfill:

confidentiality - Confidentiality means protecting the system so unauthorized access to the information is not allowed. A crucial aspect of confidentiality is user identification and authentication.

integrity - Integrity is the protection of system data and process from intentional or accidental unauthorized changes. To maintain integrity it is important to ensure that the information does not get accidentally or intentionally destroyed or modified. It is also important that threats or breaches are detected and rectified.

availability - Availability is the assurance that a computer system is accessible by authorized users whenever needed. It is important that authorized users get access to the resources when required.

Based on the goals of a security infrastructure, the OGSA infrastructure, the security issues for the specified GridIR requirements are:

4.1 authentication

Identity of the user and the services should be mutually established during any session. "User" in this context could be an individual, a group of users or another grid service.

4.2 authorization

Authorization is the procedure of determining what the user is allowed to do and it is required at various levels described below:

4.2.1 hosting environment authorization

OGSA Grid Services typically run in a hosting environment, which could be the operation system, J2EE environment or a Tomcat servlet container. Access to these environments may need to be restricted.

4.2.2 service-level authorization

There may be an authorization scheme at a coarse-grained level of the service restricting access to authorized users. The GridIR services must enforce the authorization decision as either "allow" or "deny". It is likely that a GridIR service response may be asynchronous in the case where the authorization decision may not be able to be determined immediately (for example, if a human must make an authorization decision). The GridIR service may provide additional information regarding pending decisions.

4.2.3 index-level authorization

There may be an authorization scheme to provide authorized access to a particular index. The GridIR service must enforce the authorization decision as either "allow" or "deny". It is likely that a GridIR service response may be asynchronous in the case where the authorization decision may not be able to be determined immediately (for example, if a human must make an authorization decision). The GridIR service may provide additional information regarding pending decisions.

4.2.4 record-level authorization

There may be an authorization scheme to provide authorized access to a particular record within an index. The GridIR services must enforce the authorization decision as either "allow" or "deny". It is likely that a GridIR service response may be asynchronous in the case where the authorization decision may not be able to be determined immediately (for example, if a human must make an authorization decision). The GridIR service may provide additional information regarding pending decisions. A record within an index may not be a complete data item. For example, it might be a URL pointing to a non-Grid resource. GridIR does not specify or control access to such external data items. Furthermore, GridIR does not include sub-record-level authorization ie does not restrict access based on data-items of a record.

4.3 identity management

Identities associated with services and users need to be established and maintained. Users are likely to have multiple identities based on their roles in different virtual organizations.

4.4 single sign-on

It is important the identity and credential integrates with the user's already existing local security policies. Mapping between a local identity and a Grid identity may need to be performed. The user should be allowed to access services across multiple domains using his local authentication and eliminating the need to re-authenticate.

4.5 delegation

Users may require services to access other services on his behalf. Thus, the user needs to be able to delegate his credential for such transactions.

4.6 credential renewal

In typical Grid systems, temporary or proxy credentials created from the longer-term credential of a short lifetime are used. Persistence of queries and services in GridIR requires that there be mechanisms to renew the temporary credential when it expires.

4.7 intrusion detection system

Mechanisms and monitoring practices to detect threats and breaches should be included in the security plan for the system. Administrative and system actions and policies, based on the local practices, in the event of an attack should be considered.

4.8 logging and audit

User actions and system activities need to be logged. This is vital to audit security threats and breaches. It is likely that a system may also want to use these for accounting. Policies on when logs must be retained or disposed are outside the scope of this discussion, and may need to adhere to external guidelines for data retention, privacy, or archiving.

5. Glossary

Collection - A named set of documents and associated metadata.

CQL - Common Query Language

Document - A digital entity with an associated content type and other descriptive metadata.

Index - A real or virtual entity representing (possible) data structures and methods for rapidly identifying a relevant subset of documents from a set of collections based on a query or other parameters.

IR - Information Retrieval

OGSA - Open Grid Services Architecture

VO - virtual organization

WSDL - Web Services Description Language

6. Author Information

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

Sousan Karimi
Center for Networked Information Discovery and Retrieval (CNIDR)
3021 Cornwallis Road
Research Triangle Park, North Carolina 27709-2889
email: sousan@cnidr.org

Nassib Nassar
Etymon Systems, Inc.
P.O. Box AM
Princeton, NJ 08542
email: nassar@etymon.com

Gregory Newby, PhD
Arctic Region Supercomputing Center
Fairbanks, AK 99775
email: newby@arsc.edu

7. 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 which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

8. Full Copyright Notice

Copyright (C) Global Grid Forum (date). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

9. References

[Z3950-95] Information Retrieval (Z39.50-1995): Application Service Definition and Protocol Specification, ANSI/NISO Z39.50-1995

[OGSA-PHY] Foster, F., Kesselman, C., Nick, J., Tuecke, S., The Physiology of the Grid - An Open Grid Services Architecture for Distributed Systems Integration, www.globus.org

[OGSASec] The Security Architecture for Open Grid Services and OGSA Security Roadmap. <http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/>

[WSSTec] Security in a Web Services World: A Proposed Architecture and Roadmap, <http://www-106.ibm.com/developerworks/library/ws-secmap/>

[SecurityHandbook] Tipton, H., and Krause, M. (Eds.), Information Security Management Handbook, 4th Edition, Boca Raton, FL, Auerbach, 2000.