Eui-Nam Huh, Kyung Hee University, Yongin, Korea
Pil-Woo Lee, KISTI, Daejun, Korea
Gregory B. Newby, Arctic Region Supercomputing Center
January 14, 2010

A Framework of Online Community based Expertise Information Retrieval on Grid

Status of This Document

This document provides experimental information to the Grid community regarding implementation issues for the Grid Information Retrieval research group. It does not define any standards or technical recommendations. Distribution is unlimited.

Abstract

Web-based online communities such as blogs, forums and scientific communities have become important places for people to seek and share expertise. Search engines such as Google, Yahoo!, Live etc. are not yet capable to address queries that require deep semantic understanding of the query or the document. Instead, it may be preferable to find and ask someone who has related expertise or experience on a topic. Web-based online communities are the places people often seek advice or help. Before an analysis of search capabilities for these communities can be done, we need to gather the data (questions and answers, social support or discussion, comments or advice, content rating, social relations, and so forth) that describe the communities. There is no universal standard data structure for the outline of user participation in these communities. Also, as these communities rarely interoperate, each typically only has access to its own social data and cannot benefit from other communities' data. Extracting, aggregating and analyzing data from these communities for finding experts on a single framework is a challenging task. In this document, we present a Grid-enabled framework of expertise search (GREFES) engine, which utilizes online communities as sources for experts on various topics. We suggest an open data structure called SNML (Social Network Markup Language) to outline user participation in online communities. The architecture addresses major challenges in crawling of community data and query processing by utilizing the computational power and high bandwidth inherently available in the Grid. Our framework supports open APIs for third party providers or developers to build new solutions in order to get more user feedback to improve the system.

Contents

1. Introduction

Locating the expertise necessary to solve difficult problems is a social and collaborative problem. Recent advances in Web 2.0 [1] have enabled the proliferation of many online communities and interactive collaboration spaces like wikis, forums, blogs, scientific communities and other social networking services which are gaining popularity and are enhanced with dynamic information sharing among millions of people. These communities have enabled new levels of interactions and interconnections among individuals, documents and data. They have become places for people to seek and share expertise [2].

Imagine Martin is a good Java programmer who just joined a new project for developing a program for mobile platforms. But he is new to using Java packages in mobile platforms. He is getting a warning message from his first program on the mobile platform and he is unable to locate a document explaining this message. He is not sure about whether the problem has arisen because he does not understand how to use the java package in mobile environments or because the Java package he is using does not support mobile platforms. It can be difficult to get a satisfactory answer to Martin's problem by searching Google directly. In particular, today's search engines are not yet capable of answering queries that require deep semantic

understanding of the query or the document [3]. Instead, Martin may prefer to find and ask someone who has related expertise or experience on this topic. These online communities are often ideal places for people to seek advice or help. They are bound together by shared professions, interest or products among their participants. Topics range from advice on medical treatment, programming, software, building a computer from scratch to repairing the kitchen sink and many others. This document describes work seeks to create a next generation of expert finding search framework based on these online communities.

Before analysis of search capabilities for these communities can be done, we need to gather the data (questions and answers, social support or discussion, comments or advice, content rating, social relations, and so forth) that describes online communities. Although online communities are part of the Web, their data representations are very different from general web pages. In online communities different users make use of different tools and social information is scattered. There are no standards for the outline of user participation. Also as these communities rarely interoperate, each is typically only aware of its own social data and cannot benefit from other communities' data. Thus, there is a need to define an open data structure for user participation in online communities.

There are also many challenges in crawling online community data and query processing. There is little research on the crawling of online social community data. Crawler components of search engines are responsible for locating, fetching, and storing the content residing within the online communities, while the query processor is responsible for evaluating user queries and returning results to the users relevant to their query. The efficiency problem in query processing is due to the need to quickly evaluate a query over a rather large index in the presence of many user queries being submitted concurrently [4]. In case of Web crawling, efficiency problems are due to the large scale of the Web or online communities as well as the Web's constantly evolving nature, which require pages to be downloaded and indexed frequently [5].

Another problem in case of crawling is the freshness of the collection. It is important to minimize the differences between cached data and the originals from the communities in order to update expert ranking information, thus keeping the served information up-to-date. Moreover, crawling needs a large amount of computational resources, high network bandwidth and a large amount of volatile memory to store and manage the data structures that grow quickly and continuously during the crawl [6]. Extracting, aggregating and analyzing data from these communities on a single framework is a challenging task.

In recent years, Grid computing has evolved to co-ordinate sharing of distributed and heterogeneous resources and to support a very powerful way of managing, processing and storing huge amounts of distributed data [7]. We believe that all these computational and storage requirements make web crawling a suitable target for Grid computing.

In this document, we present a Grid-enabled framework of expertise search (GREFES) engine which is different from traditional ones in both design philosophy and functionality. The framework collects, analyzes and aggregates data from different online communities or

social networks available on the Web to find people with suitable expertise on various topics. A relevance ranked list of expertise results are returned by interpreting a search string and mapping it onto related keywords. The main contributions of this document are as follows:

- A novel framework of expertise information retrieval that utilizes online virtual communities as sources of experts. It addresses major challenges in crawling and query processing in online communities by utilizing the computational power and high bandwidth inherently to the Grid.

- A data structure called SNML (Social Network Markup Language) to outline user participation in online communities.

- Supporting open APIs for third party providers or developers to build new solutions to get more user feedback to improve the system.

The rest of the document is structured as follows: Section 2 briefly reviews related works. Section 3 presents our proposed Grid-enabled expert finding search engine architecture. It also provides a description on the architectural components and features. Section 4 depicts SNML data structure and open APIs. Section 5 concludes the document with a brief summary of expected contributions and future directions.

2. Related works

In this section, we survey previous works on expert search in online communities or social networks, crawling of online community data and Grid-based search engines.

2.1 Online Communities

Online communities are used for a variety of groups interacting via the Internet for social, professional, educational or other purposes. Examples are Flickr, Facebook, Del.icio.us, Myspace, Twitter and various forums, wikis, etc. These communities have also become a supplemental form of communication between people who know each other primarily in real life. These communities do not always focused on social relationships. Instead, they reflect community member's shared interests. The ability to interact with like-minded individuals instantaneously from anywhere on the globe has considerable benefits. These communities are places for people to seek advice or help. Many have become large-scale knowledge networks which are context-dependent and multi-dimensional. User engagement and values of each community highly depends on how well it fulfill user's searches.

Typical search engines like Google and Yahoo! often fail to answer queries that require deep semantic understanding of the query or the document. But in online communities a user posts a topic or question and then some other user post replies, either to participate in the discussion or to answer a question posed in the original post. For instance, the Sun Java Forum has thousands of Java developers coming to the site to ask and answer questions related to Java programming every day. The Microsoft TechNet newsgroup is a major place

for programmers to seek help for programming questions related to Microsoft products. Usually these communities have a discussion thread structure. The reason that a user replies to a topic is typically because of an interest in the content of the topic rather than who started the thread. This also indicates that the replier has superior expertise on the subject than the asker. Thus, there is a possibility to form an expertise network [8].

2.2 Expertise Search

Expert finding systems have been explored in a series of studies, including Streeter and Lochbaum [9], Krulwich and Burkey [10], and McDonald and Ackerman [11] as well as the studies in Ackerman et al. [12]. Newer systems, which use a social network to help find people, have also been explored, most notably in Yenta [13] and ReferralWeb [14]. These systems attempt to leverage the social network within an organization or community to help find the appropriate responders. Bibliographic reference studies such as [27] and [28] aim at finding influential individuals and the evolution of co-authorship networks by analyzing conference papers of SIGMOD and SIGIR respectively.

In recent work [15], the authors proposed finding experts in an organization based on the social network. These social networks are built from two sources: from email and by extracting co-occurrences of people from web pages. An expertise propagation algorithm was proposed based on social network.

Also in [16], the authors proposed an expertise search system called Arnetminer which constructs a social network among researchers through their co-authorship and utilizes this network information as well as the individual profiles to facilitate expertise oriented search tasks. In particular, the co-authorship information is used both in ranking the expertise of individual researchers for a given topic and in searching for associations between researchers.

In [8], the authors aimed at finding experts in online help-seeking communities using social network analysis methods. They analyzed the Java forum and tested PageRank and HITS algorithms to identify users with high expertise.

The approach described in this document is different from above approaches as we are utilizing already existing online communities or social networks as sources of experts.

2.3 Crawling and Open Data Structure issues for Online Communities

There are many research studies concentrating on different issues in Web crawling, such as URL ordering for retrieving high-quality pages earlier [17], partitioning the Web for efficient multi-processor crawling [18], distributed crawling [19] and focused crawling [29]. However, there has been little documented work on the crawling of online social community data.

Online social communities are often huge, therefore crawling these networks could be both challenging and interesting. Heer and Boyd [30] mentioned their use of a crawler to gather Friendster data for their Vizster social network visualization system, but they did not go into details of the design and implementation of their crawler. Chau, Pandit et al. [31] describes

parallel crawlers for online social network utilizing a centralized queue. These examples are mostly centralized systems, not suitable for geographically distributed social network services.

Although it seems to be a simple task, there are many challenges in crawling. The two important issues are coverage and freshness. Coverage refers to the size of the set of pages retrieved within a certain period of time. A successful crawler tries to maximize its coverage in order to provide a larger, searchable collection to users. Similarly, the freshness of the collection is important in order to minimize the difference between the cached copies of pages and the originals on the Web, thus keeping the served information up-to-date. Another important issue in Web crawling is the need for a large amount of computational resources. First, a large amount of processing power is necessary to parse the crawled pages, extract hyperlinks, and index the content. Second, a large amount of volatile memory is required to store and manage the data structures that grow quickly and continuously during the crawl. The final and most important resource requirement is network bandwidth. Network bandwidth determines the page download rate and affects the crawler's coverage as well as page freshness. We believe that all of these computational requirements make crawling a suitable target for grid computing. Grids contain computationally powerful nodes, which have the resources necessary for running a crawling application. Furthermore, in cases where the spatial locality of the pages is important, the geographically distributed nature of the grid can be utilized to increase page download rates.

Online social communities are part of the Web, but their data representations are very different from general web pages. Different users make use of different tools and social information is scattered. There are no standards for the outline of user participation. In [32] authors define a universal data interchange format called DyNetML to enable exchange of rich social network data and improve compatibility of analysis and visualization tools. DyNetML is an XML-derived language that provides means to express rich social network data. Arikan and Erdogan [26] proposes an open data structure called ULML (User Labor Markup Language) to construct criteria and context for determining the value of user labor for distribution from online communities. ULML is for retrieving the statistics of content (how many photos, friends, comments, etc.) but not for actual content.

We specify a universal data structure called SNML to outline user participation in online communities. Our framework supports open APIs for third party providers or developers to build new solutions, in order to get more user feedback to improve the system. Our API is like SONAR API [29], but includes more methods and activities to satisfy our architectural needs.

2.4 Grid Enabled Search Engines

The use of the grid for information retrieval is relatively new. To the best of our knowledge, GRACE (Grid Search and Categorization) [20] and SE4SEE (Search Engine for South-East Europe) [21] are the two attempts to develop Grid-enabled search engines.

The aim of GRACE is to build a search and categorization tool over the grid. GRACE can use both local directories and the query results of other search engines as a knowledge repository. The main objective of GRACE is to analyze search results and categorize them via linguistic analysis. In this perspective, GRACE is an unsupervised categorization tool rather than a search engine. In GRACE, the utilization of grid resources is achieved via parallelism based on the distributed nature of the grid. A user can concurrently run multiple queries over the grid. GRACE, in turn, analyzes the query results, categorizes them, and aggregates the results of multiple queries.

Although GRACE and SE4SEE architectures both aim to utilize Grid resources, their motivations are different. While GRACE categorizes query results based on results obtained from other search engines, SE4SEE does not depend on the results of other search engines. Instead, query results are retrieved directly from the Web, utilizing geographical closeness in country-specific searches. Furthermore, GRACE does not provide a facility for category-specific search, whereas SE4SEE allows users to select and search in a specific category as well as perform a keyword-based search. SE4SEE does not provide real time searching, as it supports on-demand crawling.

The architecture described here deals with finding people with expertise, not documents, from existing online Web communities, by using Grid technology.

3. Framework

3.1 Architecture Overview

The GREFES infrastructure is framed by the Open Grid Services Architecture (OGSA) framework and thus relies on Service Oriented Architecture (SOA) principles and second generation Web Service standards. As the underlying grid middleware is able to distribute the work evenly, load balancing is not an issue for the current system. The framework is shown in Figure 1. The terminology used to describe the system architecture is listed in Table 1.

A user can access the search web portal using a web browser from his/her desktop or mobile devices such as laptops, mobile phones or PDAs. In order to prevent the misuse of grid resources, the user needs to register a profile to have a valid GREFES account, which is verified by the Grid security interface in the portal. The Web portal acts as a mediator between the user and the Grid. It converts the user query into a Grid job and submits it through a Grid user interface node (GUI) to a Grid worker node (GWN). GWNs are responsible for executing the crawler, expert ranking and indexing tasks. The generated expert ranking results are stored in the distributed GWNs databases. Search and resource broker (SRB) is not only coordinates the jobs and handles their assignments, but also responsible for submits the search query to OGSA-DAI (Open Grid Service Architecture-Data Access and Integration) after getting appropriate content source from CSS (Content Source Selection) service. OGSA-DAI deliver the ranking results to web portal to satisfy user query.

We can see that our architecture has two parts: a processing part (social network data extraction, expert ranking and indexing) and an expert searching part. We will present them in detail in section 3.2 and section 3.3.

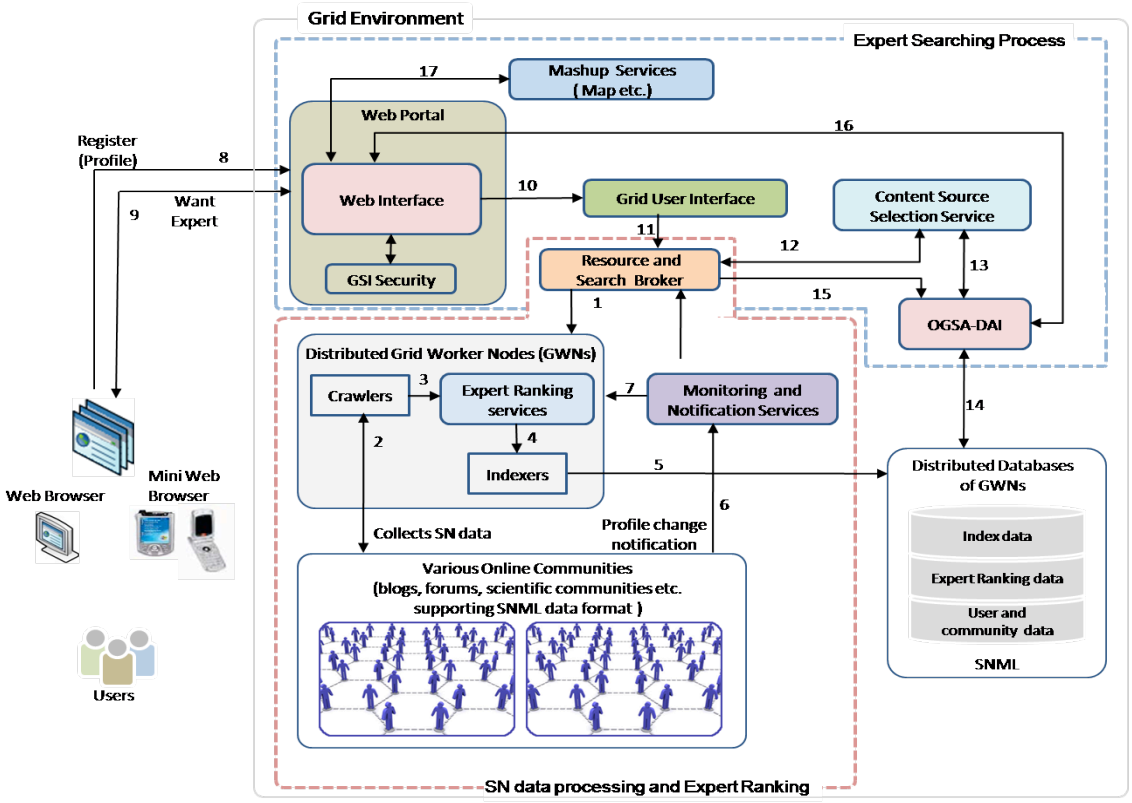

Figure 1: Framework of expertise information retrieval on the Grid

Table 1: List of commonly used terms

Terminology	*Description*
Crawler	Locating, fetching, and storing the content residing within the online communities
Indexer	The expert data is transformed to expert indexes by indexers
GWN	Grid worker nodes (GWN) are distributed computing resources which execute crawling, expert ranking and indexing tasks
SNML	Social Network Markup Language (SNML) is an open data structure to outline user participation in online communities
ERS	Expert ranking service (ERS) ranks experts on various topics using a ranking algorithm
MTEN	Monitoring and event notification (MTEN) monitors Grid worker nodes and notify them if any profile change in online communities
Web Portal	Web interface for searching expert
GSI security	Grid security interface (GSI) to validate registered user
SRB	Search and resource broker (SRB) coordinates the jobs and handles their assignments to worker nodes. Also sends search query to OGSA-DAI
GUI	Grid jobs are submits to SRB through Grid user interface (GUI)
CSS	Content source selection (CSS) generates and maintains content source descriptions and optimizes the distribution of queries by selecting suitable sources

OGSA-DAI	Open Grid Service Architecture-Data Access and Integration (OGSA-DAI) enables connection to distributed databases and deliver results to the web portal
Mashup Service	For example, Google map to display user location

3.2 Processing Framework

The processing framework is designed for collecting social network data from online communities and for ranking experts on various topics. The major components of the processing parts are shown in Figure 2 and explained as follows:

Crawler – Since the GREFES framework deals with expertise search, intended to serve a large number of users each with specific, personal expert needs, a crawler is used to collect SNML files (described in detail in section 4). These files contain user activities (log data – comments or advice) data, profile data (may include expertise areas) and relation data from online communities. In order to be able to adapt to the heterogeneous nature of Grid infrastructure, a platform independent crawler should be preferred which is capable of executing on different architectures, thus avoiding recompilation overhead and compatibility issues. Many crawler threads can execute concurrently on GWNs in order to overlap network operations with CPU processing, thus increasing throughput.

The crawlers in Grid worker nodes retrieve community data in a breadth-first fashion. A queue data structure is used to store the list of pending users which had been seen but not crawled. Initially, a seed set of users was inserted into the queue. Then at each step, the first entry of the queue is popped, all information for that user was crawled, and every user left (but who was not yet seen) was queued. Once all of the user's information was crawled, the user was marked as visited, and stored in a separate queue.

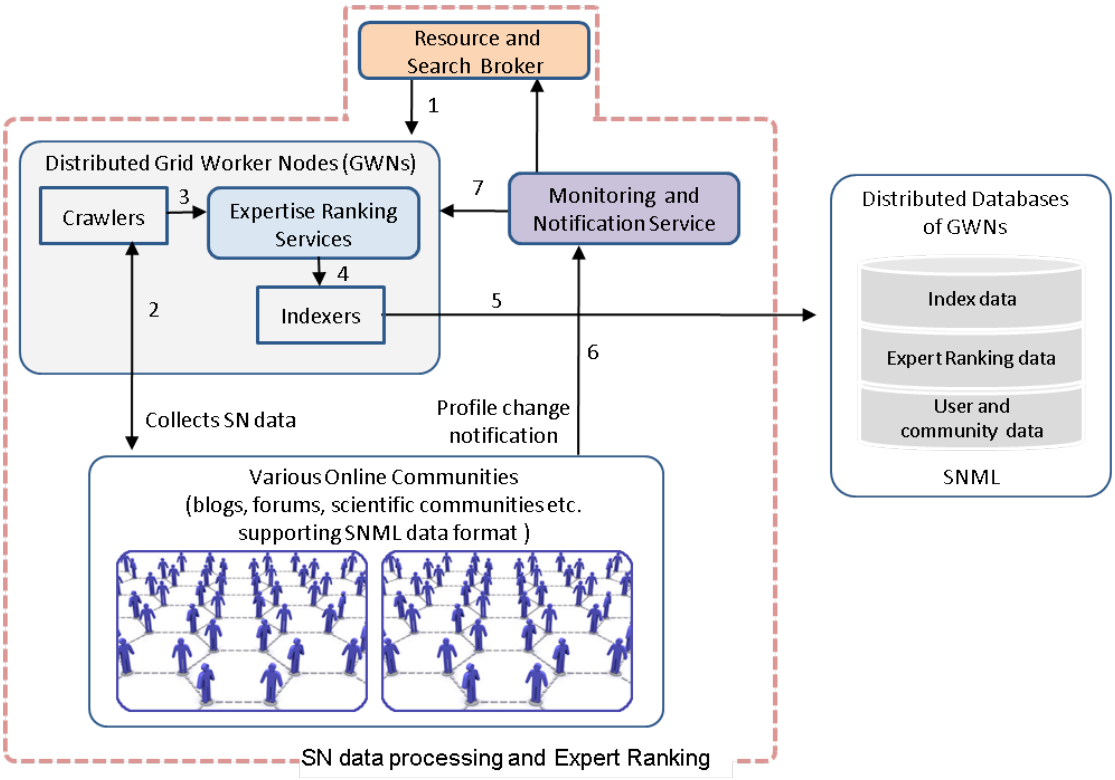

Figure 2: Social network data processing and expert ranking framework workflow

Expert Ranking Service (ERS) – The main goal of the search engine is to find suitable experts on a given topic. So after collecting SNML files of users through crawlers from different communities, an ERS service is used to find users' knowledge or expert areas (if not mentioned by users) and to calculate their expert ranking values on the areas using a ranking algorithm. Finally, a relevance ranked list of experts is presented to the user. If ranking values are same for experts on a topic from different communities, a community rank value is used for final ranking of experts. The community which has more members will be ranked more highly. We will provide more detail about the expert ranking algorithms in section 5.

Indexer – Expert ranking results are indexed by indexers and stored in the distributed worker nodes databases. This helps to quickly find suitable expertise on various topics and thereby reduce query response time.

Search and Resource Broker (SRB) – It coordinates and schedules the jobs on worker nodes and also responsible to send search query to OGSA-DAI by contacting content source selection (CSS) service.

Monitoring and Event Notification (MTEN) Service – This monitors distributed worker nodes and notifies them if any profile update is available in the online communities. Upon notification, the resource broker schedules worker nodes to execute crawlers.

The workflow as shown in Figure 2 is explained step by step as follows;

Step 1*: Resource broker coordinates the jobs of distributed worker nodes. A worker node runs multiple crawler threads concurrently for collecting social network data from online communities.*

Step 2: *After collecting and processing SN data, expert ranking service is performed in the worker node using a ranking algorithm to find experts on various topics in different online communities.*

Step 3*: The indexer in the worker node creates expert ranking indexes for efficient query processing, and stores them in worker node databases.*

Step 4*: If any profile changes in online communities, MTEN service notifies a worker node which then updates the expert ranking results.*

3.3 Expert Searching Process Framework

This framework provides a relevance-ranked list of experts to a user on a topic of interest. The major components of this framework are shown in Figure 3 and are explained as follows:

Web Portal – This is the only interaction point between the user and the GREFES back-end, and thus a major component of the architecture. Users register their profiles as well as submit their queries through this portal. Queries are submitted as a Grid job through Grid user interface (GUI). The portal has to be user-friendly, even though it requires a more complex interface than classic search engines due to the application's added capabilities such as mashup services (i.e. Google map etc.).

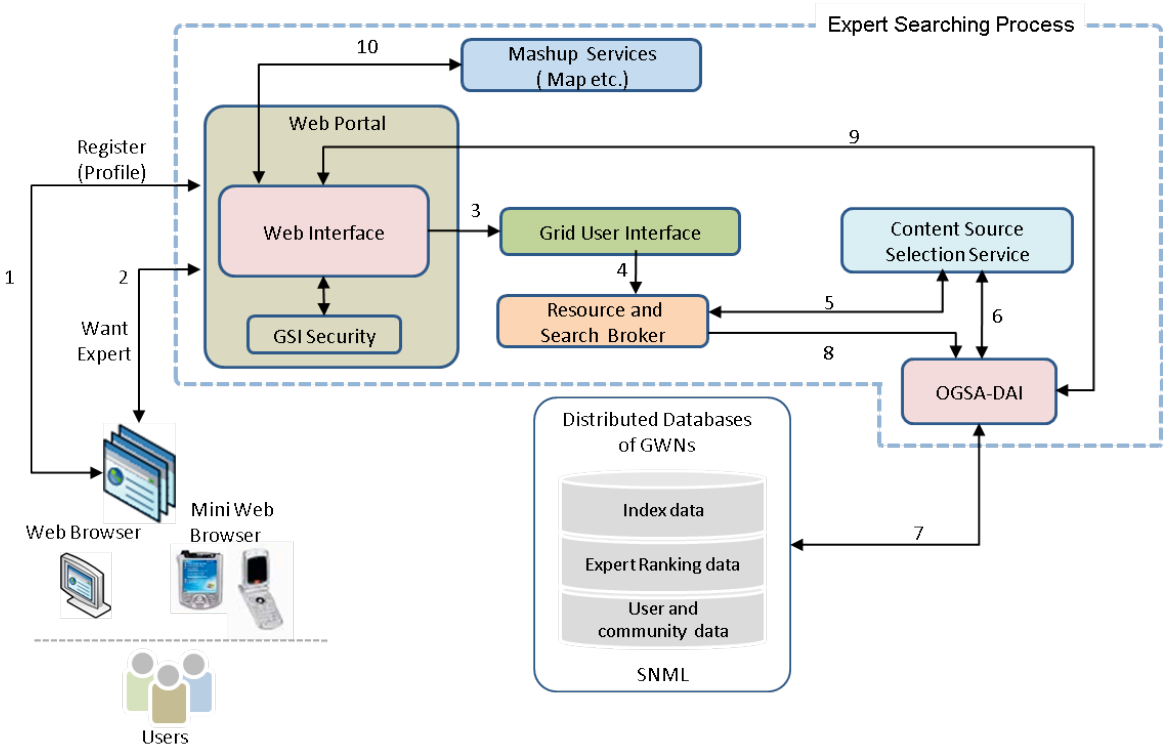

Figure 3: Expert search process framework workflow

Content Source Selection (CSS) Service – This generates and maintains expert content source descriptions, which include partial content indices, summative content indices, or result traces from training or past queries. It uses CORI [22] selection algorithm (the standard collection selection mechanism) and Language Modeling techniques [23] to find appropriate expert index collections. Thus it optimizes the distribution of queries by selecting target expert index sources which are relevant to a given query.

OGSA-DAI – OGSA-DAI [24] is an extensible framework for data access and integration. It exposes heterogeneous data resources to be accessed via stateful web services. Advantages of OGSA-DAI are indicated as follows

- No additional code is required to connect to a database or for querying data. OGSA-DAI supports an interface integrating various databases such as XML databases and relational databases.

- OGSA-DAI provides three basic activities -- querying data, transforming data, and delivering the results using ftp, e-mail, and push services.

- It also allows users to develop additional activities. This aspect supports scalability in the system, and presents powerful means to use data resources in various ways.

In the architecture, OGSA-DAI is used to query expert indexes from distributed worker nodes databases and then to deliver query results to the web portal.

The workflow of the expert search process is shown in Figure 4 and is explained step by step as follows:

The first stage of the process is the warm up sequences denoted in the Figure 4 as steps (1)-(6), with steps (7) – (13) denoting the querying sequence.

Step 1: The SRB contacts the CSS to initialize the service to collect index descriptions.

Step 2: The CSS contacts the OGSA-DAI service to acquire a description of each index.

Step 3: OGSA-DAI requests descriptions from the index.

Step 4 and 5: The index provides collection statistics in a pre-defined format to the CSS through OGSA-DAI.

Step 6: Steps (2)-(5) are repeated for each index, until the all indexes are described, and the warm up of the CSS is complete.

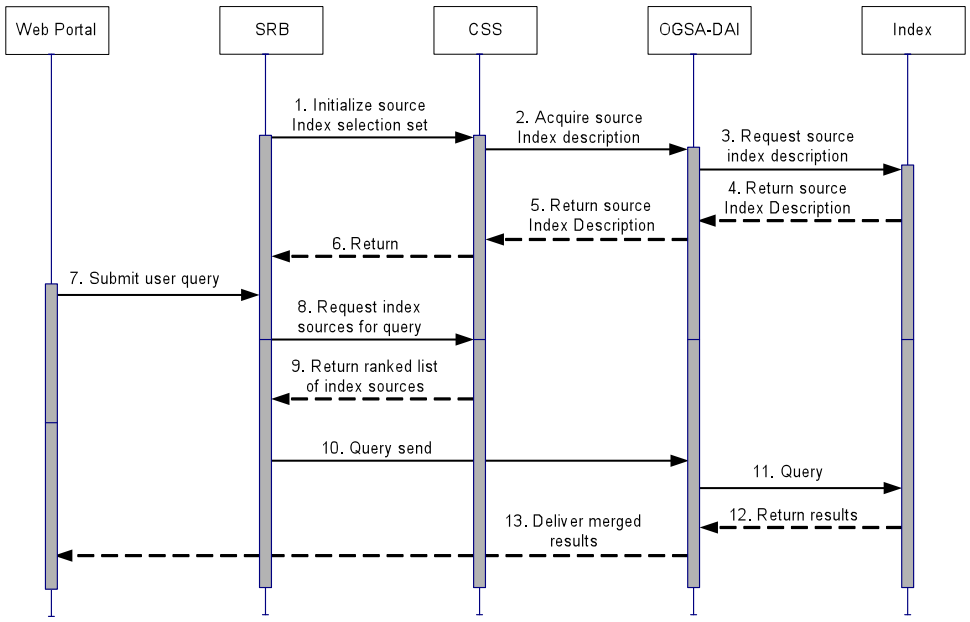

Figure 4: Sequence diagram of the search process. Steps (1) - (6) denote the warm-up sequence, and steps (7) - (13) denote the query sequence.

Step 8: The query is sent to the CSS for selection of the set of indexes.

Step 9: The CSS returns a ranked list of indexes. This may be the top n *indexes which are most likely to satisfy the query, or all those indexes scoring above a certain threshold. These criteria can be defined when performing the selection.*

Step 10 and 11: The selected indexes are then contacted through OGSA-DAI and queried for finding relevant experts on a topic. This process is performed concurrently for efficiency (via threading).

Step 12 and 13: The expert ranking result sets, produced by each index from distributed worker nodes databases, are merged. Final ranking results of experts are delivered to the web portal through OGSA-DAI.

3.4 Architectural Features

The distinguishing features of this framework are as follows:

1) Utilizing online Web communities as sources of experts

2) Using an open API to gather and share expertise data from online communities

3) Alleviating the computational burden of Web crawling by the utilization of resources available in the Grid.

4) Running queries against multiple distributed collections with a likelihood of possessing relevant sources in parallel, thus eliminating the need to search unnecessary collections.

5) Rather than seeking sub-second response times over billions of documents, as monolithic search engines do, the system seeks fast performance over a far smaller number of collections. This will enable more complex query processing.

6) Because of their smaller size and locality to a particular collection source, the distributed collections can be updated more quickly, thus eliminating the delay among harvest runs exhibited by search engines.

4. SNML (Social Network Markup Language) and Open APIs

SNML is an open data structure protocol based on XML that can be generated by the service provider and follows the framework described here. Crawlers collect SNML documents, which describe community data in a universal format. SNML documents contain community information, user's participation information such as experiences, expertise areas, activities, relations, etc. SNML consist of four sections: as shown in Figure 5 and described briefly as follows:

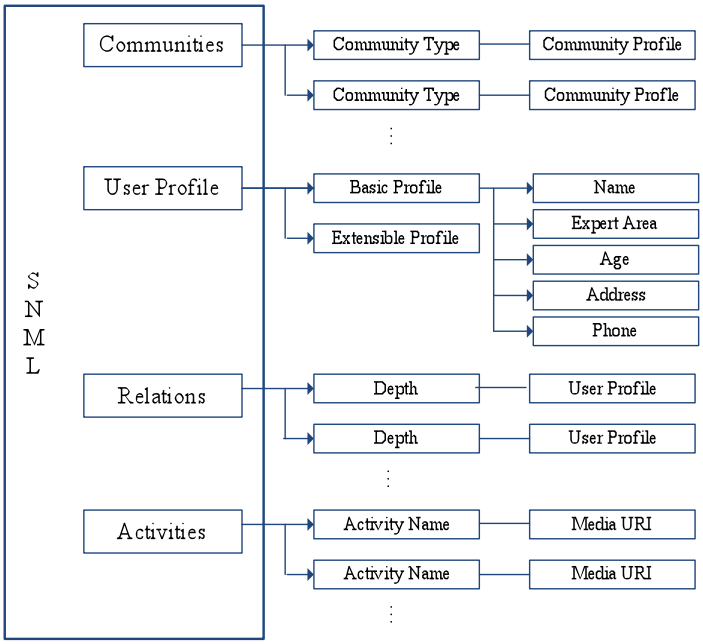

Figure 5: SNML sections.

<Community>: The community a user belongs to. It consists of community type and community profile. Community type describes type of medium, such as blog, wikis, forums, etc., and community profile describes the community activities or services.

<User Profile>: Consists of a basic profile and an extensible profile. Basic profile includes personal information such as name, age, expertise area, phone, address, etc. Extensible profile depends on different social network or communities. For example, Facebook's profile data includes information other than the basic profile.

<Relation>: Consists of user profiles connected to the current user and their depth. It may include the number of friends connected to the user, their profiles, etc.

<Activities>: Consists of an activity name and media URI (Universal Resource Identifier). It includes user comments, posts, etc.

The SNML data format is shown in Figure 6. Besides SNML, this architecture supports open APIs that are like REST APIs. Our system, as well as third party providers or developers, can use these APIs to develop new solutions. Such new developments can provide experiences and user feedback to improve the system.

```
<? xml version="1.0"?>

<SNML xmlns="urn:GIR:2008:08-SNML-NS"
    xmlns:SNML="urn:GIR:2008:08-SNML-NS">

<community mimeType="text/plain">
                    <type/>
                    <profile/>
</community>

<profile mimeType="text/plain">
                    <basic>
                              <name/>
                              <expertise area>
                                        <item/>
                                        <descriptor/>
                              </expertise area>
                              <age/>
                              <address/>
                              <phone/>
                    </basic>
                    <extension>
                              <!-- extensible profile  -->
                    </extension>
</profile>

  <relation mimeType="text/plain">
                    <depth/>
                    <profile>
          <!-- user profile based on profile in SNML  -->
                    </profile>
</relation>

<activities mimeType="text/plain">
                    <name/>
                    <uri/>
</activities>

</SNML>
```

Figure 6: SNML data structure format

The three APIs provide useful method for communication. They are summarized below:

Profile API: This API helps to easily find people, their relationships and strength of relationships. Methods are summarized in Table 2.

Community API: This API is used to understand the community site or user group. Methods are summarized in Table 3.

Activities API: Users in communities perform various activities. We can capture these activities through this API. Methods are summarized in Table 4.

Table 2: Profile API methods

Resource	*URI*	*Output*
User Resource	/profile/{userid}	<user profile in SNML>
Rank Resource	/profile/{userid}/rank	Float(0.000 to 10.000)
Network Resource	/profile/{userid}/network/limit	<graph of people in SNML>
Relation Resource	/profile/{userid}/relation/depth/limit	<list of people in SNML>

Table 3: Community API methods

Resource	*URI*	*Output*
Service Resource	/community/{communityid}/service	<list of web services in XML>
Site Rank Resource	/community/{communityid}/rank	Float(0.000 to 10.000)
Site RSS Resource	/community/{communityid}/rss	<RSS feed>

Table 4: Activity API methods

Resource	*URI*	*Output*
Activities Resource	/activity/{userid}/activities/limit	<list of Activities in SNML>
Rating Resource (Optional)	/activity/{userid}/rating	Float(0.000 to 10.000)
Activities RSS	/activity/{userid}/rss	<RSS feed>

5. Security Considerations

This document presents a description and API for community expertise recommender systems. We identify two security considerations. First, there is a consideration of accuracy in the information retrieval techniques utilized. For example, a system might present inaccurate or misleading ratings. This security consideration is out of scope for the framework, but is important for implementers.

A second area for security consideration has to do with implementation of the API. The proposed API does not include data integrity tests for data at rest, or in transmission (i.e., to avoid unintentionally merging records), or other features for robustness. To address these concerns, it is assumed that any implementation of the API would be based on a robust framework or toolkit that provides data integrity, authentication, authorization, and other features – as well as methods for data exchange and record encoding. These are seen as necessary for implementation, but out of scope for the current document.

6. Conclusion

In this document, we have presented a new framework for expertise information retrieval on computational Grids. It utilizes online web communities as sources of experts on various topics. The framework specifies an open data structure called SNML for sharing community data efficiently and effectively. The architecture addresses major challenges in crawling online community data and query processing by utilizing the computational power and high bandwidth available in the Grid. Several open APIs are described so that people can build new solutions utilizing the framework.

7. Contributors

Eui-Nam Huh
Kyung Hee University
Korea
Johnhuh@khu.ac.kr

Pill-Woo Lee
KISTI
Korea
pwlee@kisti.re.kr

Gregory B. Newby
Arctic Region Supercomputing Center
Alaska, USA
newby@arsc.edu

Seung-Min Han
Kyung Hee University
Korea
han905@khu.ac.kr

Mohammad Mehedi Hassan
Kyung Hee University
Korea
hassan@khu.ac.kr

8. Intellectual Property Statement

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 OGF Secretariat.

The OGF 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 OGF Executive Director.

9. Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF 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.

10. Full Copyright Notice

11. References

[1] T. O'Reilly. *What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software,* 2005

[2] J. Zhang and M. S. Ackerman. *Searching For Expertise in Social Networks: A simulation of Potential Strategies.* ACM GROUP'05, Nov, 2005, USA

[3] J. Bian, Y. Liu, E. Agichtein and H. Zha. *Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media*. ACM WWW 2008, April 21-25, Beujing, China

[4] B. B. Cambazoglu and C. Aykanat. *Performance of query processing implementations in ranking-based text retrieval systems using inverted indices.* Information Processing & Management, 42(4), 875–898, 2006

[5] B. B. Cambazoglu, Turk and C. Aykanat. *Data-parallel Web crawling models.* Lecture Notes in Computer Science, 3280, 801–809, 2004

[6] J. Cho and H. Garcia-Molina *The evolution of the Web and implications for an incremental crawler.* In Proceedings of the 26th international conference on very large data bases (pp. 200–209). 2000, Cairo, Egypt

[7] I. Foster and C. Kesselman. *The grid 2: Blueprint for a new computing infrastructure.* San Francisco: Morgan Kaufman, 2003

[8] J. Zhang, M. S. Ackerman and L. Adamic. *Expertise Networks in online Communities: Structure and Algorithms. WWW 2007, May 8-12, Banff, Alberta, Canada*

[9] L. Streeter and K. Lochbaum. *Who Knows: A System Based on Automatic Representation of Semantic Structure*. In *Proceedings of RIAO*, 1988, 380-388

[10] B. Krulwich. and C. Burkey. *ContactFinder agent: answering bulletin board questions with referrals*. In the 13th National Conference on Artificial Intelligence, Portland, OR, 1996, 10-15

[11] D. W. McDonald and M. S. Ackerman. Expertise Recommender: A Flexible Recommendation Architecture. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW '00)*, 2000, 231-240.

[12] M. S. Ackerman, V. Wulf and V. Pipek, (eds.). *Sharing Expertise: Beyond Knowledge Management*. MIT Press, 2002.

[13] L. N. Foner, Yenta: a multi-agent, referral-based matchmaking system. In *Proceedings of Agents '97*, ACM Press, Marina del Rey, CA, 1997, 301-307

[14] H. Kautz, B. Selman. and M. Shah. *Referral Web: combining social networks and collaborative filtering.* Commun. ACM, 40 (3). 63-65

[15] Y. Fu, R. Xiang, Y. Liu, M. Zhang and S. Ma. *Finding Experts Using Social Network Analysis. IEEE/WIC/ACM Intl. Conf .on Web Intellligence, 2007*

[16] J. Li, J. Tang, J. Zhang, Q. Luo, Y. Liu,(eds.) *EOS: Expertise Oriented Search Using Social Networks.* WWW 2007, May 8–12, 2007, Banff, Alberta, Canada

[17] R. Baeza-Yates, C. Castillo, M. Marin and A. Rodriguez. *Crawling a country: better strategies than breadth-first for Web page ordering.* In Special interest tracks and posters of the 14th international conference on World Wide Web, 2005, Chiba, Japan.

[18] P. Boldi, B. Codenotti, M. Santini, and S. Vigna. *Ubicrawler: a scalable fully distributed Web crawler.* In Proceedings of AusWeb 2002, the eighth Australian World Wide Web conference.

[19] B. B. Cambazoglu, A. Turk, and C. Aykanat. *Data-parallel Web crawling models*. Lecture Notes in Computer Science, 3280, 801–809.Wide Web Conference 2004 (pp. 161–172). Brisbane, Australia.

[20] F. Scholze, G. Haya, J. Vigen, and P. Prazak. *Project GRACE: a grid based search tool for the global digital library.* In 7th international conference on electronic theses and dissertations. Lexington, KY, 2004

[21] B. B. Cambazoglu, E. Karaca, T. Kucukyilmaz, A. Turk and C. Aykanat. *Architecture of a Grid Enabled Web Search Engine.* Journal of Information Processing and Management 43 (2007) 609-623, Elsevier

[22] J. P. Callan, Z. Lu, and W. B. Croft. *Searching distributed collections with inference networks*. In SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pages 21–28, 1995

[23] L. Si, R. Jin, J. Callan, and P. Ogilvie. *A language modeling framework for resource selection and results merging*. In CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, pages 391– 397, 2002.

[24] M. Antonioletti et al. *Web services data access and integration – The core (WS-DAI) specification, version 1*. GFD-R.74. Lamont, Illinois: Open Grid Forum.

[25] I. Guy, M. Jacovi, E. Shahar S. Farrell et al. *Harvesting with SONAR- The Value of Aggregating Social Network Information.* ACM CHI 2008, April 5-10, 2008, Florence, Italy

[26] B. Arikan and E. Erdogan *A framework for sustaining user labor across the web* .2008. www.userlabor.org

[27] M. A.Nascimento, J. Sander, J. Pound. *Analysis of SIGMOD's coauthorship graph.* ACMSIGMOD Record, 2003, 32(3): 8–10

[28] A. F. Smeaton, G. Keogh, C. Gurrin, et al. Analysis of papers from twenty-five years of SIGIR conferences: what have we been doing for the last quarter of a century. ACM SIGIR Forum, 2002, 36(2): 39–43

[29] I. S. Altingovde, and O. Ulusoy. *Exploiting interclass rules for focused crawling. I*EEE Intelligent Systems, 200419(6), 66–73.

[30] J. Heer, and D. Boyd. *Vizster: Visualizing Online Social Networks.* IEEE Symposium on Information Visualization (InfoVis), 2005

[31] D. H. Chau, S. Pandit et al. *Parallel Crawling for Online Social Networks.* ACM WWW 2007, May 8-12, Banff, Alberta, Canada

[32] M. Tsvetovat, J. Reminga and Kathleen Carley. *DyNetML: A Robust Interchange Language for Rich Social Network Data.* Institute for Software Research, International Carnegie Mellon University, 2005