PROCEEDINGS OF SCIENCE

Interactive Information Extraction based on Distributed Data Management for German Grid Projects

René Jäkel*
Center for Information Services and High Performance Computing, Technische Universität Dresden
E-mail: rene.jaekel@tu-dresden.de

Steffen Metzger
Max-Planck-Institute for Informatics
E-mail: smetzger@mpi-inf.mpg.de

Jason Milad Daivandy
Jülich Supercomputing Centre
E-mail: j.daivandy@fz-juelich.de

Katja Hose
Max-Planck-Institute for Informatics
E-mail: hose@mpi-inf.mpg.de

Denis Hünich
Center for Information Services and High Performance Computing, Technische Universität Dresden
E-mail: denis.huenich@tu-dresden.de

Ralf Schenkel
Saarland University
E-mail: schenkel@mmci.uni-saarland.de

Bernd Schuller
Jülich Supercomputing Centre
E-mail: b.schuller@fz-juelich.de

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence.

PoS(EGICF12-EMITC2)031

PROCEEDINGS OF SCIENCE

The current infrastructure proviced and maintained by the German Grid Initiative (D-Grid) primarily covers resource management and exchange at the data level supporting mainly technical resources such as computational capacity, data transport networks, storage resources, and management software. The WisNetGrid project (www.wisnetgrid.org) aims to broaden the focus of resource sharing towards the actual content, such as research and production data, to enable interdisciplinary usage. To achieve this goal, resource sharing is supported on different abstraction layers. First, we create an information layer by providing a universal interface to access data on the grid independent of the underlying grid storage system. Second, at the knowledge layer, we offer interactive knowledge extraction and management tools that can also take advantage of a community's grid resources. These tools enable the user to formulate the domain specific knowledge in different ways to ease the interaction with the knowledge extraction process and to provide input for automatic extraction workflow. Within this project, we work together with use groups from the humanities and from landscaping as disparate use cases to evaluate which advantages can be gained by using semi-automatic extraction tools to gather and manage knowledge content.

PoS(EGICF12-EMITC2)031

EGI Community Forum 2012 / EMI Second Technical Conference
26-30 March, 2012
Munich, Germany

*Speaker.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence.

1. Introduction

In the past years, grid computing [1] has evolved for many different disciplines to solve not only computationally intense tasks, but also to provide resources for data management or the provision of services to steer applications, manage users, or create complex workflows. A strong advantage of grid computing is the ability to maintain distributed and heterogeneous resources using a middleware layer to distribute computational jobs or delegate user access to data storage space. This provides a unique way to handle data transparently to the user community not knowing the technical details of the underlying infrastructure. This approach works well for a rather uniform and straightforward user community with similar needs and requirements. The picture changes for scientific approaches where collaborative work among partners from different domains is desired with a broader range of requirements in terms of used software or data and knowledge handling. The rather static grid approach using an individual middleware solution for a project might introduce additional hurdles for new participating use groups not familiar with grid computing techniques, while still demanding the ability to handle a broader range of different data types and applications.

The WisNetGrid project[1] from the German grid initiative D-Grid[2] aims to overcome these limitations by supporting resource sharing on different abstraction levels. To enable transparent access to data resources, we establish a uniform information layer on top of widely used grid middlewares and other resource types providing access to distributed heterogeneous data. This enables a knowledge layer providing services for knowledge generation and management on top of the information layer by applying information extraction methods to gather semantic knowledge from available information provided by the underlying storage systems. Typical domain-independent information extraction methods, which go beyond named entity recognition, are either imprecise or computationally expensive, and even the most precise systems need human supervision for critical applications. Additionally, extraction methods often work iteratively. Thus, detecting mistakes early has a significant positive impact on the performance and can reduce the overall need for human corrections. Our goal is to provide extraction services that tightly integrate user feedback into the extraction process while minimizing human effort needed to obtain optimal results. To achieve this goal, an interactive extraction system is provided, which learns during the extraction process from user feedback. This allows the extraction tools to implicitly learn domain specific knowledge about how to extract knowledge, which reduces the initial human effort to adapt the system for a particular knowledge domain. Our approach also enables the use of grid middlewares to transfer computationally intensive tasks to the grid.

Within the project we evaluate our approach on concrete example scenarios in collaboration with two different science communities. One such scenario originates from a humanities project[3], which focusses on studying the works of famous writers. While our federation layer can enable the integration of data from different sources, our extraction methods can help to relate either the works of an author themselves, or particular locations, persons and other named entities mentioned in such works to historical facts from a general knowledge base or other sources.

[1] The project homepage is addressable via: http://www.wisnetgrid.org

[2] D-Grid - German Grid Initiative: http://www.d-grid.de

[3] TextGrid - Virtual Research Environment for the Humanities: http://www.textgrid.de

PoS(EGICF12-EMITC2)031

In this article, we first introduce the data management developed within the project WisNet-Grid in Section 2. Based on this data access Section 3 presents the interactive knowledge extraction process from a user's point of view and discusses interactive possibilities for use groups to provide their domain-specific knowledge while making use of the extraction methods provided to enlarge their knowledge base.

2. Distributed Data Management

In many scientific fields data from different resources have to be considered for research and analysis, or are created by a broad range of applications. Usually, communities create and manage their data in a sense that allows for a most efficient workflow in terms of providing input for community specific applications and storing results for further processing and analysis. The data format is usually highly driven by the processing methods and the kind and amount of data which has to be processed.

In practice many different reasons determine the use of specific technical solutions to address and manage storage resources for various use groups. Those storage systems might be databases, web-based free or restricted data repositories, filesystems, and storage systems provided by cloud or grid middlewares. To be considered as basis for collaborative working, and to be able to combine data from various systems, different access mechanisms have to be followed , either by passing required username/password or a certificate to the storage system to authenticate the user. The data handling might also be dependent from data type and storage solution and also mostly incompatible between different data back-ends.

In the grid context, the middleware layer addresses heterogeneous computing resources and communicates with the underlying resources. It provides ways to access data, mechanisms for job submission and steering and the management of users and access to grid services. Usually, the services of different middlewares are not compatible with each other. To use various data sources managed by different middleware solutions, or in general different data management systems, different data access mechanisms have to be combined in a uniform manner.

2.1 Related activities and projects

In conjunction with collaborative efforts, several approaches have been introduced to overcome these difficulties. The European Grid Infrastructure (EGI)[4] aims to provide a federation infrastructure for grid providers and their users as basis to provide secure access to federated computational and storage resources [2]. This way, Virtual Research Communities (VRC) shall be supported in their domain-specific development to participate from developments in other research areas, in particular by incorporating new grid ready applications and profit from communication with a broader international community.

Within this context, the European Middleware Initiative (EMI)[5] is a collaboration of four middleware providers (ARC, dCache, gLite, UNICORE) to incorporate more standards to increase the interoperability between individual middlewares to be more attractive for a broader range of

[4]European Grid Infrastructure: http://www.egi.eu

[5]European Middleware Initiative: http://www.eu-emi.eu

PoS(EGICF12-EMITC2)031

communities. If the effort and complexity of deploying and operating grid infrastructures can be decreased, the initiation of grid computing to new users and scientific fields could be eased.

Furthermore, by extending existing grid infrastructures towards techniques from cloud computing [3] could offer other ways of usage and access to new or occasional users, e.g. within a temporarily research project for partners not from the established scientific community. Some projects currently investigating those extensions toward cloud computing to evaluate the efforts to open the grid computing domain towards new usages. The community project Stratuslab develops an open-source based cloud distribution to allow grid and non-grid resource providers to offer IaaS-cloud[6] solutions and evaluate this approach for different use cases from the molecular and structural biology and bio-informatics [4]. This cloud-based extension is an interesting aspect to new users, since the user has more freedom in configuring the guest system in a IaaS-cloud infrastructure. Nonetheless, by still using traditional grid infrastructure extended by cloud computing aspects might just shift the technical hurdle for new users towards different technical aspects. Furthermore, compared to the grid the cloud approach is not yet open for the federation of resources across different domains [2] and presents itself to the user as a rather closed system.

Another aspect to lower the hurdle for new research groups accessing Distributed Computing Infrastructures (DCI), or in particular grid infrastructures, is by the provision of portal-based interaction mechanisms for users. To name just a few examples, projects like PGRADE, GridSphere, or MosGrid as example community from the D-Grid initiative, addressing this issue for different scientific areas. Using a portal service allows the interaction with the underlying computing system in a common sense with a focus on the specific application the user is interested in [5], but new user groups have to rely on a strong community which initially provides well established applications running on DCI's.

2.2 Resource Federation Layer

To provide a common access to underlying data resources e.g. as basis for higher information extraction services, we introduce a federation system, which provides uniform access using the WebDAV protocol[7]. WebDAV is an extension of the HTTP/1.1-protocol[8] and allows to enable routines for locking and copying/moving of data or meta data content. This hides the individual management systems from the user and provides uniform operation to handle data and metadata and allows a representation of the content with a unique URI. Since there are many different ways to store and represent data the interaction between the federation layer and the actual data management system has to be realized by a specialized connector. A general picture about the basic structure of the federation system is illustrated in Figure 1.

The resource federation component realizes the routing from the requesting client to the storage system and provides a uniform message format. The data or metadata content of the connected resources are presented to the user via its URI within the WisNetGrid namespace. The data federation component selects the relevant connector to establish the connection to the resource. The connector can be seen as interface between data or meta data sources and the federation layer and

[6] Infrastructure As A Service

[7] Web Distributed Authoring and Versioning (WebDAV): specification online as HTTP extensions via http://www.webdav.org/specs/rfc2518.html

[8] The specification can be found online: http://www.w3.org/Protocols/rfc2616/rfc2616.html

PoS(EGICF12-EMITC2)031

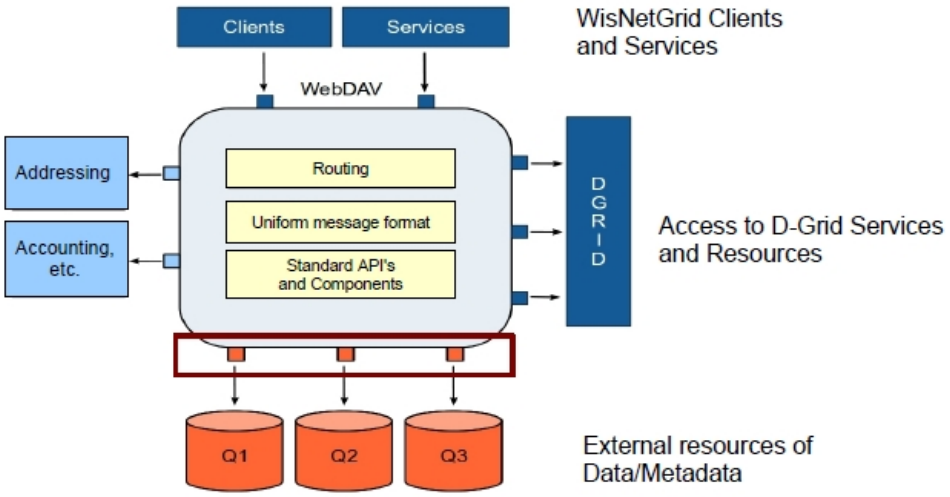

Figure 1: Structure of the resource federation layer. Highlighted (red box) are the connectors to the external resources.)

also provides basic functionality for manipulating them (CRUD). It takes the request from the user, performs the login to the data management system, processes the request, and converts the data to a WebDAV-conform format. This allows not only to access the data by WisNetGrid services, but also to browse the data via any HTTP/1.1-capable client for which the user has access to.

The resource federation component also provides a mechanism to enable the user to manage the credentials needed to access distributed storage systems. This is the basis to realize a real session management to provide a Single Sign-On (SSO) mechanism to delegate the user request to the data management system. Once the session is started the user is authorized to access data resources for which he has appropriate credentials provided. By using WisNetGrid services, like the information extraction workflow, the system can act on behalf of the user to delegate the access request to the underlying data management system by invoking agents as part of the SSO infrastructure.

3. Interactive Information Extraction

The WisNetGrid project aims at providing a generic framework supporting two levels of knowledge extraction. First, the typical task of named entity recognition (NER) is supported. Traditionally, this task is defined as the identification of referenced entity types, i.e., identifying occurrences of the strings *"Berlin"* and *"Frankfurt"* as references to places, occurrences like *"Goethe"* and *"German Chancellor"* as references to persons, etc. However, recent work in this field allows for uniquely identifying individual entities referenced in texts of a broad domain, given some background knowledge about the domain entities [6, 7]. This allows for identifying *"Goethe"* in a text about German literature as a reference to the particular person named Johann Wolfgang von Goethe and *"Frankfurt"* in a text about that writer as a reference to Frankfurt near the river Main, where Goethe was born, instead of the smaller Frankfurt on the Oder. The problem to decide which actual entity among a set of potentially addressed entity candidates is the one actually meant in the text is known as *disambiguation*. Based on such a named entity recognition on the level of individuals,

PoS(EGICF12-EMITC2)031

relations connecting recognized entities can be extracted from texts as well. Such relations might, for instance, be the birthplace of persons, the books a writer authored, or the movies an actor acted in. While there are different approaches [8, 9, 10], WisNetGrid applies an iterative pattern-based approach that aims at extracting instances of a fixed set of predefined binary relations. A pattern in the most abstract sense is a recurring construct, e.g. a word phrasing or a tabular representation, expressing the abstract relations textually. For instance, from the text *"Goethe, who was born in Frankfurt, is one of the most famous German writers."* an instance of an abstract `wasBornIn` relation can be extracted linking Johann Wolfgang von Goethe to his birthplace Frankfurt on the Main. This abstraction can only be made if the extraction system already *knows* that the textual pattern *"X, who was born in Y"* is a representation of that particular relation.

The system we use is an extension of the pattern based SOFIE extraction framework [10, 11, 12]. It learns iteratively which textual patterns express which abstract relations from relation instance examples given. In parallel, additional relation instances are derived in each iteration using the patterns previously learnt. This iterative approach has the advantage that the actual patterns and their meaning can be learnt based on examples, such that a user does not need to understand the technical details of the system. To provide a relevant basis for the relation extraction process, the user just needs to provide example instances of the relations to be extracted along with text snippets containing a textual representation of these example instances. The more patterns can be learnt from the examples, the more new relation instances can be derived from the text to be used as additional examples. During in the next iterations more and more patterns can therefore be used in order to extract more relation instances to refine the results and so on. Although this method can achieve good results [10], it may make mistakes. Due to the iterative nature of the process, it is imperative to reduce such mistakes as early as possible to improve the overall quality and minimize the human effort needed to correct wrongly recognized relation instances. Furthermore, the system needs to learn which textual representations can reference to which abstract entities. An automatic system simply does not know by itself which entities could be represented by the term *"German Chancellor"* without someone telling it who the current and all the previous office holders are. Such specific information, which we call domain-specific knowledge, can only be given by an expert in the particular field for which text-based input data is to be processed.

3.1 Work-flow and implicit domain knowledge gathering

The domain knowledge mainly consists of the actual entities of interest, their possible reference names and example instances of relations between such entities. Additionally, the system relies on a type system based on a class taxonomy. Each entity has at least one and potentially many types assigned from the class hierarchy, e.g. Johann Wolfgang von Goethe might be associated with the types `Person`, `Writer` and `GermanWriter`. Similarly, relations are type-casted, i.e. each relation has a domain and range type. For instance, the `wasBornIn` relation could have a domain of type `Person` and a range of type `Place`. In order to make it as easy as possible for end-users to provide the necessary domain knowledge, and give direct feedback early on, the system integrates a feedback loop into the extraction process. While some domain specific background knowledge, as the set of interesting entities and the related typing information, needs to be provided a priori, the system can also learn interactively from user feedback, without a need for a user to understand the technicalities behind. Once a certain set of basic domain knowledge is provided from user side

PoS(EGICF12-EMITC2)031

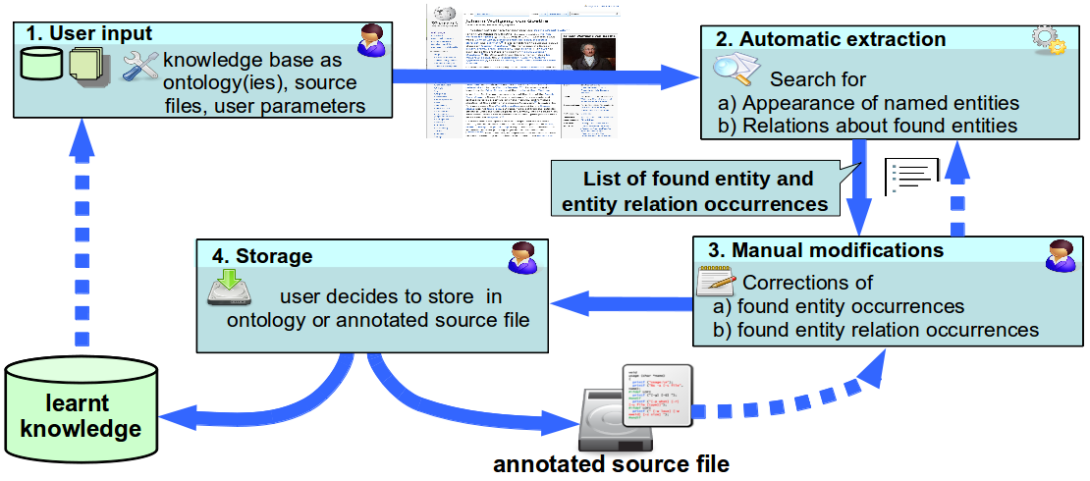

Figure 2: Interactive extraction workflow.

a typical workflow would follow the illustration in Figure 2. First a set of files is selected along with the correct background knowledge settings for the domain of these files. Then the automatic extraction system processes these files. Afterwards a user can inspect the recognized entity and relation instance occurrences and if necessary correct 1) which entity is referenced, 2) which relation is expressed between two entities, 3) add new entity references or 4) add new relation instances. In each case the system implicitly learns from these corrections and this knowledge is applied to the whole file or set of files when the extraction is re-run. Thus, a user needs, for instance, only once indicate that *"Frankfurt"* in a text does indeed reference *the other Frankfurt* near the Oder, and the engine will get it right in the whole text. Similarly pattern quality, i.e. the probability that a pattern always expresses a certain relation, can be automatically adjusted, when a user flags a statement as a wrongly or correctly identified relation instance. Finally, results can be exported once a user is satisfied.

3.2 Integration into the WisNetGrid architecture

For the interaction with the extraction system a simple web-based user interface is provided on top of a web-service API allowing the implementation of more sophisticated front-ends, e.g. enabling integration into a particular workflow environment of any community. Thus, following the general WisNetGrid approach to provide unified access to resources, the end-user interaction is clearly separated from the technical details of the extraction system in place, Given the interface is supported the extraction system could simply be replaced by any other such system. In addition, the extraction system is of course integrated with the general WisNetGrid architecture, such that file access, for instance, uses the federation layer. That means in particular the source files are accessed via WebDAV-links in the common WisNetGrid namespace and access authentication is handled by the federation layer independently from the extraction system, as described in the previous section. Similarly, available computation resources can also be accessed via the federated resource layer in order to distribute the extraction computation onto a community's grid infrastructure. For instance, the initial parsing of all files to be considered in the extraction process can easily be spread onto

PoS(EGICF12-EMITC2)031

several machines, as recognition of potential entity references and patterns is independent of other documents. To achieve this we have used the UNICORE adapter to formulate extraction jobs using the same extraction tools. Only later when patterns are interpreted information from several source documents needs to be considered, yet this step can be split by the patterns investigated.

3.3 Example scenario application

In our humanities oriented application scenario, the research community can first integrate their data, i.e., the digitized works of authors of interest, into the WisNetGrid data federation in order to benefit from the unified access. This could be done by using an adapter to a storage system or by defining links to public data sets, such as Wikipedia. Once the data is accessible and some domain knowledge has been provided in ontological form, a researcher can use the interactive extraction system to identify entity (and fact) occurrences of different form in the given data set. In our example scenario mentioned in the beginning of this paper this could be historical text sources from books or articles available in a dedicated repository[9]. This allows, for instance, to trace the appearance of a particular entity in different contexts, e.g., linking different names for persons and thus allowing for easier analysis of a book's or an entity's relevance in a historical context as well as to investigate an entity's or factual statement's changing perception throughout history or in different text versions. Furthermore, this extracted knowledge can be used in a more general manner for further analysis or simply be stored using the uniform data access layer for later use of this particular group.

4. Conclusion and Outlook

We provide tools to enable content oriented resource sharing on two levels. First on the data level, by providing uniform access to different grid-based data storage back-ends. Second on the knowledge level, by providing interactive grid-based information extraction methods. Our information extraction methods can work automatically, given adequate domain knowledge, yet they can also incorporate human feedback directly into the iterative extraction process, thus learning globally by human feedback. As large scale information extraction is computationally very expensive, our architecture allows to use grid resources for computational tasks, if such grid resources are available to the user community, while preserving the user's control over the extraction process. We evaluate our architecture with concrete usage scenarios of different research communities to ensure a generic framework usable by a broad audience. However, this also means some components need domain specific adaptations to achieve their best performance in a particular setting. Additionally, the area of information extraction is still a very active field, such that further significant advancements are to be expected in the near future. However, we assume our framework to provide interfaces generic enough to allow relatively easy integration of future developments and other extraction methods.

[9]For this we have integrated the data repository from the German project TextGrid.

PoS(EGICF12-EMITC2)031

References

[1] I. Foster and C. Kesselman (Eds), *The Grid: Blueprint for a New Computing Infrastructure*, 2nd Edition, Morgan-Kaufmann, 2004

[2] A. Di Meglio, *Grids and Clouds Integration and Interoperability: an overview*, PoS(ISGC 2011 & OGF 31)112, 2011

[3] I. Foster, Zhao Yong, I. Raicu and S. Lu, *Cloud Computing and Grid Computing 360-Degree Compared*, in Grid Computing Environments Workshop, 2008. GCE '08, pages 1–10, 2008

[4] StratusLab: Enhancing Grid Infrastructures with Virtualization and Cloud Technologies, http://stratuslab.eu/index.php

[5] Peter Kacsuk, *P-GRADE portal family for grid infrastructures*, Concurrency and Computation: Practice and Experience, Volume 23, Issue 3, pages 235–245, 2011

[6] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan and Soumen Chakrabarti, *Collective annotation of Wikipedia entities in web text*, in *KDD*, pages 457–466, 2009

[7] Mohamed Amir Yosef, Johannes Hoffart, Marc Spaniol and Gerhard Weikum, *AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables*, in Proceedings of the 37th International Conference on Very Large Data Bases, volume 4, pages 1450–1453, 2011

[8] S. Auer and C. Bizer and G. Kobilarov and J. Lehmann and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data*, In ISWC/ASWC, pages 11–15, 2007

[9] Sergey Brin, *Extracting Patterns and Relations from the World Wide Web*, In WebDB, pages 172–183, 1999

[10] Fabian M. Suchanek, Mauro Sozio and Gerhard Weikum, *SOFIE: A Self-Organizing Framework for Information Extraction*, in WWW, 2009

[11] Ndapandula Nakashole, Martin Theobald Gerhard Weikum, *Scalable Knowledge Harvesting with High Precision and High Recall*, in *WSDM*, pages 227–236, 2011

[12] Shady Elbassuoni, Katja Hose, Steffen Metzger and Ralf Schenkel, *ROXXI: Reviving witness dOcuments to eXplore eXtracted Information*, Proceedings of the 36th International Conference on Very Large Data Bases, volume 3, pages 1589–1592, 2010

PoS(EGICF12-EMITC2)031