Graph-based Data Integration in EUDAT Data Infrastructure

Vasily Bunakov*, Paolo D'Onorio De Meo†, Stephan Kindermann‡, Anna Queralt§ and Jedrzej Rybicki¶
*Science and Technology Facilities Council (STFC),
Harwell Campus, Didcot OX11 0QX, UK
Email: vasily.bunakov@stfc.ac.uk
†CINECA,
Via dei Tizii 6, 00185 Rome, Italy
Email: p.donoriodemeo@cineca.it
‡Deutsches Klimarechenzentrum GmbH (DKRZ),
Bundesstrasse 45a, 20146 Hamburg, Germany
Email: kindermann@dkrz.de
§Barcelona Supercomputing Center (BSC),
Carrer de Jordi Girona 29, 08034 Barcelona, Spain
Email: anna.queralt@bsc.es
¶Juelich Supercomputing Center (JSC),
Wilhelm-Johnen-Strasse, 52425 Juelich, Germany
Email: j.rybicki@fz-juelich.de

***Abstract*—European Data Infrastructure (EUDAT) is a distributed research infrastructure offering generic data management services to the research communities. The services deal with different phases of the data life cycle, some of them are tailored to account for special needs of the individual communities or replicated to increase the availability and resilience. All that leads to scattering of the large and heterogeneous data across service landscape limiting discoverability, openness, and data reuse. In this paper, we show how graph database technology can be leveraged to integrated the data across service boundaries. Such an integration will facilitate better cooperation among the researchers, improve searching and increase the openness of the infrastructure. We report on our work in progress, to show how better user experience and enhancement of the services can be achieved by using graph algorithms.**

Keywords–Data Integration; Graph Databases; Designing for Open Data; Linked Data.

I. Introduction

Nowadays, it is widely accepted that public data, and in particular research results, should be made accessible to society, facilitating better, more efficient science and innovation. In line with the open data and open access movements, EUDAT [1] is a pan-European initiative building a sustainable cross-disciplinary and cross-national data infrastructure providing a set of shared services for accessing and preserving research data. The EUDAT services work with digital collections comprised of data objects. The term data object in EUDAT is pretty broad and encompasses structured data, text, multimedia binaries, binary output of scientific simulations, and much more. In this paper, we will use terms data object, digital object, and object interchangeably. EUDATs vision is to enable European researchers and practitioners from any research discipline to preserve, find, access, and process data in a trusted environment, as part of a Collaborative Data Infrastructure (CDI).

The problem with a generic infrastructure like EUDAT is that it must fulfill a lot of expectations at the same time. The expectations come from different communities or usage scenarios. The usual way of dealing with different community requirements is to add new services to the infrastructure portfolio or tailor the existing ones accordingly. It is a strength and weakness at the same moment. The cost of the flexibility is the complexity of the service landscape. It is further amplified by the geographical distribution used to increase the scalability and resilience of the infrastructure. There are many instances of the same service created at different locations to serve different groups of users. Users use different services to tackle different problems or phases of data life-cycle. Altogether, this leads to fragmentation of the content: some data objects are uploaded to one service, others to other service. In extreme cases, it can even happen that the same data object is uploaded to many services as there is no way of finding out if and where it was previously stored. External identifiers as used by some services, for instance in form of handles (like [2]), do not necessary help. They are opaque, hash-based values generated independently of the content of the object. To cope with this heterogeneity a much more expressive model of the data stored in the infrastructure is required.

In computer science, every decent software design starts with an analysis of the domain model [3]. This approach is not directly applicable to the EUDATs case. The reason for that is the heterogeneity the project has to deal with. As a resource and service provider it is not in the position to define a common domain model to account for all the special use cases originating from the communities. It rather tries to account for the domain models coming from different communities and map them on services in generic CDI. In this paper, we show how we provide the communities with a unified view of the infrastructure and the data that are already stored in the existing EUDAT services. Such integrated view will enable better understanding of the data, make a first step towards data interoperability, increasing openness, and potentially facilitate data reuse. We show how we plan to establish and store such integrated model of the different data sources, and how it allows for new features and service extensions.

It is good to offer tailored services to attract users but it is at least equally important to use content collected in the infrastructure as an attractor. Researchers can be interested in using the CDI solely based on the content it stores. The abundance of content might lead to a situation where it is hard to find or even be aware of all the data objects relevant for given scientific endeavor. The challenge, which is not unique to EUDAT, is to make the collected content visible, and searchable in ways going far beyond the currently supported keyword-based searches or faceted searches. Application of graph-based algorithms [4] revolutionized the way the Internet search engines work and how people engage in social interactions [5]. We believe that such algorithms might not be directly applicable for the scientific communities and data (e. g., most popular data set might not be the most attractive for the researchers). It would be, however, beneficial to offer graph-based descriptions of the content so that individual users can work on their own searching algorithms or just explore the content in an interactive way. The graph abstraction is already used to successfully tackle Big Data challenges [6].

Our goal is to create a generic infrastructure service to integrate the content gathered from different sources. As a service provider we are not in a position to impose a common domain model on all the communities we serve. Therefore, we provide a flexible service to describe single use cases or domains as interactive graphs. This is an abstraction that is well tested, easy understandable and quite powerful at the same time.

The rest of the paper is structured as follows. We present our design in Section II. We proceed with a short description of the implementation approach. Subsequently, an overview of the use cases currently worked on is given. We conclude this work with a summary and a list of future challenges in Section V.

II. Design

The core services offered by EUDAT CDI are shown in Figure 1. B2DROP is a service for storing, synchronizing, and exchanging dynamic research data with colleagues or team members. B2SHARE provides an easy way to upload, tag and share research data, which is made citable via persistent identifiers (PID). B2SAFE enables an automatic, rule, and policy-driven replication of data across a federation of data centres. B2STAGE allows data to be staged into and out of the CDI to, for instance, external high-performance computing services to process the data. Finally, B2FIND exposes a metadata catalog through a user-friendly, web-based search portal and a standard API. The authentication and authorization infrastructure (AAI) is orthogonal to all these services, and controls access to the infrastructure.

To improve the discoverability of the content scattered across different services and locations, we aim at providing a unified, expressive view of all the data items collected. We decided to include relations between objects to add flexibility to the model and allow for exploration of the content by just following those links. In other words, we create a graph describing the infrastructure and integrating the content collected across many services.

A valid approach for data integration and an often prerequisite for further analysis are so called “data lakes”. They are collections populated by the data extracted from all the

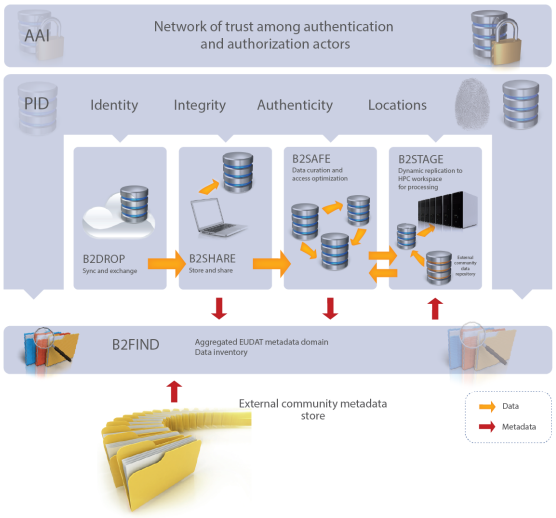

Figure 1. EUDAT Service Landscape

services in an infrastructure. Although the approach is valid it is also controversial. Especially the need for replicating the data might render it prohibitively expensive. Therefore, we decided not to duplicate the content but just include the metadata representation of data objects. In our graph, we model them as nodes with properties describing details like object name, creation date, etc. Graph nodes are also used to model further entities like service instances, people, or metadata objects. To model all kinds of dependencies between digital objects we use relations (edges in graph).

When tackling data integration one can follow bottom-up or top-down approach. In case of bottom-up, the data are gathered from services with help of specialized spiders, cleansed (if required), and then uploaded to a common repository to provide complementary, integrated view of the content. Top-down approach, on the other hand, promotes the repository to the single user-facing service with just one view of the data. During the upload of the data, the individual users describe the object with help of graph semantics. From there, the data are propagated to individual back end services. Both approaches have their advantages and drawbacks. Since we are still in an exploratory phase of implementing the service, we decided to follow the bottom-up approach: Gather as much data as possible, provide alternative view of the infrastructure and data, evaluate the benefits of such data integration and (in case of positive result) promote the service. At least for some services also an intermediate step would be possible: Graph database could be used to substitute the existing relational back end.

The bottom-up approach produces graphs describing domains of single services or domains of single communities. In the process of data integration, those graphs shall be merged together. To this end, integration points (graph overlaps) have to be identified. In general, there are two kinds of graph overlaps: common nodes in two or more graphs and relations connecting nodes originating from different graphs. An obvious candidate for a common node is a person: the same user can own data objects across multiple services. Also, metadata nodes describing people like affiliation, community, or research

interests can constitute good integration points. Another type of graph overlaps are the digital objects. It is, for instance, possible to have replicas of an object stored in different places or a set of objects derived from a given root object. Some EUDAT services assign external persistent identifiers to the managed object, so this could be clearly used to identify the same object across services. As stated above we are not storing the actual content of the digital objects, thus it is not possible to define content-based identity of any given objects. We do, however, store metadata describing objects. This metadata are either technical metadata (like checksum), or community-provided semantic metadata like provenance description or keywords. Some of the metadata will create common nodes across services but metadata can be used to identify similar objects across service boundaries. Such a similarity can be modeled as a relation (graph edge) crossing service boundaries. In the future, we plan to incorporate new services for extracting even more features from digital objects and store those features in the common graph. This can be based for instance on Linked Data AppStore [7] and would certainly help to identify commonalities between different domains.

The high-level goal of our design is "about making links, so that a person or machine can explore the web of data", which is a quote from the seminal Tim Berners-Lees note on Linked Data [8]. We try to incorporate as many good design principles from the world of linked data as possible. There are, however, some implementation details which differ from the usual way in which linked data is implemented. First of all, some of the services in EUDAT CDI do not offer HTTP(S) URIs for accessing the data. Secondly, we are in sought of benefits from exploring the graph and applying graph algorithms. Therefore, we decided not to use the RDF [9] end point but rather upload the data to a graph database where people can interact with it. In other words, we squashed together the steps of collecting the data and exploring the data. In the future we can expose the collected data as RDF and SPARQL interface to account for more sophisticated use cases and enable better integration with other infrastructures.

III. Implementation

In this section, we describe some of the implementation details of our work in progress on graph-based data integration. As already explained we follow a bottom-up approach and in the first step extract data from different EUDAT services to create distinct graph models in those bounded contexts. In the next step we integrate the data by connecting the single graphs. To manage graphs we use graph database neo4j [10], a native graph database available under GPLv3 license. It supports full Atomicity, Consistency, Isolation, Durability (ACID) consistency model, and it offers an interactive graphical interface, clients in many different programming languages, and a ReST API, leaving us with many options with regard to integration with other services as well as offering access to end users. neo4j uses property graph as internal graph model. It means that nodes in the graph can have properties and each node can be labeled with (multiple) labels. Labels can be used to divide the entities in the graph into different "abstraction classes". Properties, on the other hand, describe particular entities. Relations in property graph can have properties and names, they are also directed. neo4j offers quite a flexibility with respect to properties. It is not required that all the nodes

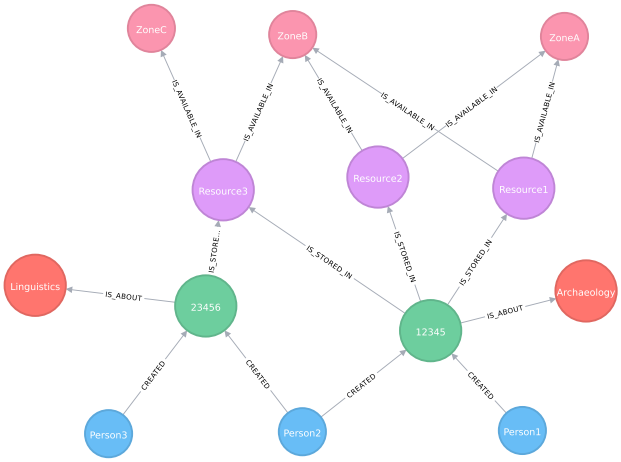

Figure 2. Example of graph data integration

have the same properties. Even if they share the same label it is still possible to introduce the heterogeneity. An example would be a graph with nodes representing people where some of the nodes have a property called address (if people decided to share their address) while others do not. This kind of flexibility in the graph model is really useful for evolving the model over time, e. g., when new features are added or more data are collected we can simply add properties to newly created nodes while keeping old nodes valid and potentially update them later. This kind of evolution is proven to be hard in relational databases. For an extensible explanation and comparison of current graph data models we refer the reader to [11].

IV. Use cases

This section describes the use cases we are currently implementing to showcase the advantages and possibilities that arise from the graph representation of the EUDATs data.

B2FIND stores metadata about the digital objects, including their authors, language, and discipline, among others. By structuring this information as a graph, it is easy to infer new relationships from the already existing ones. For instance, if two persons are authors of the same object, then we can assume that they know each other. And if a person recently created objects belonging to a discipline, we can infer that this person works in this discipline. This information can be used, for instance, to look for collaborators with a particular expertise, as well as how to reach them through co-authorship relationships by means of a shortest path query. We plan to further integrate this information with the data coming from the authentication and authorization service, which also stores affiliation of the users. In this way, we can restrict the searches, for instance by finding only experts from a given institution or country. A clear application of this use case is to propel collaboration between researchers based on the identified social-network-like links. But also more technical benefits can be obtained, for instance the location of the data object can be changed based on the expected usage to optimize the access times.

A rather more technical than social use case is fed with the data from the B2SAFE service. There the data objects are registered, replicated to avoid data lost, and made referenceable via globally unique persistent identifiers managed by the corresponding administrative domains. The PID can be also used to locate the copies (replica) of data objects across

different federations. A graph database is used to model the ownership of the data, actual replication paths of data objects, and store technical metadata describing the objects. The model also include collections (with metadata descriptions) to extend the limited functionality of the actual B2SAFE back end. There are at least two benefits of this data. First of all, it gives the data owner a good view of the infrastructure and status of their data and thus improve the trust in the CDI. Secondly, the graph database could relate a person to all of its PID and replicas across all federations allowing for better data accessibility.

Both aforementioned use cases can be used to understand the actual data integration we are sought after. Let us consider a graph as the one shown in Figure 2, where the digital objects, identified by their persistent identifiers (12345 and 23456), are gathered from the different EUDAT services. The resources in which an object is replicated, as well as the zones in which a resource is available, are obtained from the B2SAFE service. The authors of a digital object and the discipline related to it can be collected from B2FIND.

The B2SHARE use case is pretty close to the B2SAFE case. There are, however, two important differences. The content in B2SHARE is currently not replicated and there are community-provided metadata descriptions available. The model we are currently using to store this information is pretty straight-forward. For each data object we have an uploader, set of metadata (currently modeled as a single graph node) and set of keywords (each keyword is a separate node). An interesting application of this model is to provide the users with their individual "universe" composed of data objects, keywords, and people. The universe is generated with a breadth-first search of given depth and includes the "most important" objects from the domain. This feature can be incorporated into B2SHARE in the future, combined with the social-like features described in the first use case.

A different kind of use case is implemented by a representative of European Network for Earth System Modeling (ENES), which is one of the EUDAT communities. This climate research community is developing comprehensive Earth system models capable of simulating natural climate variability and human-induced climate change. The use case concentrates on modeling the distributed ENES data federation: the organization of datasets in collections served by data services hosted by data servers. The services come both from ENES community and EUDAT. The sole existence of such an overview contributes to better mutual understanding between a community (ENES) and provider of generic services (EUDAT), potentially resulting in a better usage of services. Since the model includes information about data objects harvested from EUDAT and ENES worldwide data collections, there are some interesting overlaps. EUDAT cataloged data collections are from a later phase of the data life cycle (published archived objects with a DOI assigned). The data are still worked on, thus newer versions of the same collections (or subparts of the collections) are accessible in the ENES data federation. By connecting those two worlds an integrated view of a life cycle of a digital object can be derived and the provenance of single objects and collections can be better tracked and understood.

Finally, the semantic annotations service developed in EUDAT, called B2NOTE is yet another use case for the graph database integration. The goal of B2NOTE is to provide a plug-in to the graphic interfaces of other EUDAT services for human annotation, as well as text mining tools for the automated annotation in the back end. So, the graph database could successfully address and handle the annotation provenance records: who, when, in what EUDAT service and by what tool or machine agent has produced the annotation. The annotations will be then available across all services.

V. Conclusion and Future Work

In this paper, we reported on our work in progress showing how the graph database technology allows EUDAT to integrate the data across services boundaries to provide features originally missing. We are still in a preliminary phase but the first use cases implementation already lead to some improvements, for example an easier way to walk through relations inside and across the separate services. We design our experiment in such a way that the main focus was laid on the use cases and not on, e. g., selecting the best graph database or best service and data integration scenario. This later subjects remain open until the potential of the approach is positively verified.

In our work, we have identified some challenges. Some of them will follow from our decision not to clone content inside the graph database, but only include metadata. We will have to provide a means to keep the metadata up-to-date and efficiently retrieve the data from all services from all federations. On the higher abstraction layer, we will have to work on identifying integration points between services. Such points will have to be non-intrusive, as we are not going to impose anything on the data models of single services nor communities. To this end, more effort will be made in the data cleansing process. Lastly, although this is not yet a formal requirement, we are considering an offering of an RDF endpoint, for instance, to facilitate data exchange with other infrastructures.

Among the most promising improvements identified so far, is the potential to offer better user experience by making the borders between services less visible and less relevant to the users. After a successful integration of the data, all the information will be available in all the services. The better user experience is also given by the possibility to add social-network-like features to the existing services and offer links to explore the data domain of EUDAT. In particular, a much more powerful and customizable searching functionality can be implemented based on the data collected in the graph database. Finally, the integrated information can be used to better understand the community domains, access patterns and use cases, and the EUDAT CDI itself to tune it accordingly.

Acknowledgment

The work has been supported by EUDAT2020, funded by the European Union under the Horizon 2020 programme - DG CONNECT e-Infrastructures (Contract No. 654065).

References

[1] W. Gentzsch, D. Lecarpentier, and P. Wittenburg, "Big data in science and the EUDAT project," in SRII Global Conference, Apr. 2014, pp. 191–194.

[2] R. Kahn and R. Wilensky, "A framework for distributed digital object services," International Journal on Digital Libraries, vol. 6, no. 2, Apr. 2006, pp. 115–123.

[3] E. Evans, Domain-Driven Design. Addison-Wesley, 2004.

[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems, vol. 30, no. 1-7, Jul. 1998, pp. 107–117.

[5] J. Weaver and P. Tarjan, "Facebook Linked Data via the Graph API," Semantic Web – Interoperability, Usability, Applicability, vol. 4, no. 3, 2013, pp. 245–250.

[6] V. N. Gudivada, S. Jothilakshmi, and D. Rao, "Data management issues in big data applications," in ALLDATA 15: The 1st International Conference on Big Data, Small Data, Linked Data and Open Data, Apr. 2015, pp. 16–21.

[7] R. Dumitru et al., "The linked data AppStore," in Mining Intelligence and Knowledge Exploration, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8891, pp. 382–396.

[8] T. Berners-Lee. Linked data. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html [retrieved: Dec., 2015]

[9] G. Schreiber and Y. Raimond. RDF 1.1 primer. [Online]. Available: http://www.w3.org/TR/rdf11-primer/ [retrieved: Dec., 2015]

[10] J. Webber, "A programmatic introduction to Neo4j," in SPLASH '12: 3rd ACM Annual Conference on Systems, Programming, and Applications: Software for Humanity, Oct. 2012, pp. 217–218.

[11] R. Angles, "A comparison of current graph database models," in ICDEW '12: 28th IEEE International Conference on Engineering Workshops, Apr. 2012, pp. 171–177.