iRap – An Interest-Based RDF Update Propagation Framework

Kemele M. Endris, Sidra Faisal, Fabrizio Orlandi, Sören Auer, Simon Scerri

University of Bonn & Fraunhofer IAIS, Bonn, Germany
{endris,faisals,orlandi,auer,scerri}@cs.uni-bonn.de

Abstract. We present the iRap (interest-based RDF update propagation) framework that propagates only relevant parts of updates published by a source dataset to a local replica excerpt of the source dataset. This interest-based update propagation enables remote applications to 'subscribe' to relevant datasets and consistently reflect the necessary changes locally without the need to frequently replace the entire dataset (or a relevant subset).

1 Introduction

Providing services on top of large LOD datasets is becoming a challenge due to the lack of service levels regarding availability of the datasets and restrictions imposed by the publisher on the type of query forms and number of results [3]. Hosting a replica of large datasets, such as DBpedia and LinkedGeoData, is costly, but organizations might host only relevant (interesting) parts of those datasets using approaches such as RDFSlice [2]. Once the sliced dataset is hosted, the source dataset continues to evolve in terms of content and ontology. Maintaining a consistent and up-to-date replica of the relevant data is a major challenge. Resources in the source might be added, updated or removed from the dataset. In this poster paper, we introduce the iRap – interest-based RDF update propagation framework - an implementation for *interest-based RDF update propagation approach* presented in [1]. iRap enables organizations to update their local replicas by registering interest query patterns and propagate changes from source datasets regularly.

2 Approach

The iRap framework[1], is implemented in Java using Jena-ARQ[2], consists of three modules: (1) Interest Manager (IM), (2) Changeset Manager (CM) and (3) Interest Evaluator (IE), each of which can be extended to accommodate new or improved functionality.

[1] http://eis.iai.uni-bonn.de/Projects/iRap.html

[2] https://jena.apache.org/

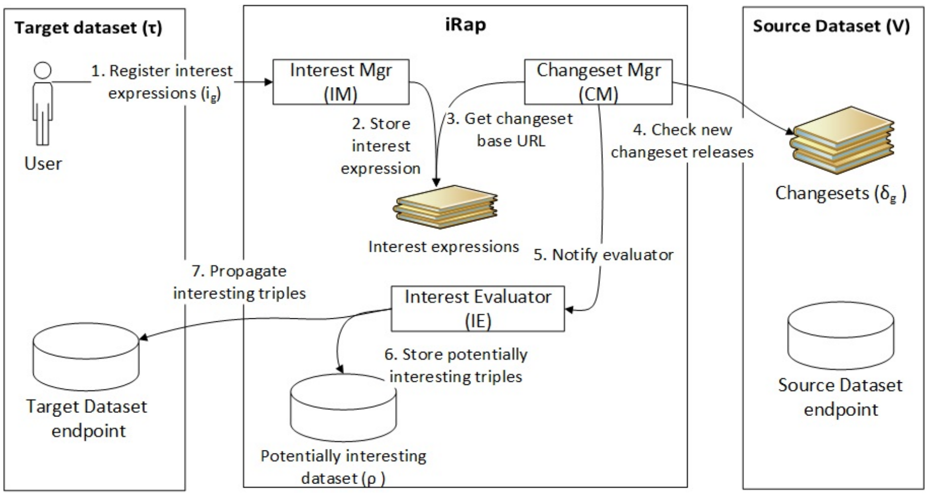
Target dataset (τ)
iRap
Source Dataset (V)
User
1. Register interest expressions (i_g)
Interest Mgr (IM)
Changeset Mgr (CM)
2. Store interest expression
3. Get changeset base URL
4. Check new changeset releases
Changesets (δ_g)
Interest expressions
5. Notify evaluator
7. Propagate interesting triples
Interest Evaluator (IE)
6. Store potentially interesting triples
Target Dataset endpoint
Source Dataset endpoint
Potentially interesting dataset (ρ)

Fig. 1: iRap – Interest-based RDF update propagation framework

The IM component is responsible for managing subscriptions, i.e, registration of interest expressions. An organization provides information about the source dataset URI to synchronize with, an interest query to select relevant parts of a changeset, and a target dataset endpoint (or Jena TDB) URI to propagate interesting updates, as shown in Listing 1.1. iRap evaluates interest expressions over changesets being published along with the source dataset. The CM module regularly checks for new changesets from the source dataset URI. After downloading and decompressing the new changesets, the CM notifies the IE, which then imports a list of registered interest expressions and initiates the evaluation. The result of executing an interest evaluation of an interest expression over a changeset are: *(1) interesting, (2) potentially interesting, and (3) uninteresting triples.*

Interesting triples are triples that fully match the interest query, but not necessarily the optional graph patterns (OGPs). Potentially interesting triples are triples that match only parts of the basic graph pattern (BGP) of an interest query. Uninteresting triples are those triples that do not match with any of triple patterns of an interest query. Interesting and potentially interesting triples originating from the set of removed triples are called interesting-removed triples and potentially interesting-removed triples, respectively. Interesting and potentially interesting triples originating from the set of added triples are called interesting-added triples and potentially interesting-added triples, respectively. The evaluation process can be summarized as follows:

Step 0 The user configures a target dataset (local replica)
Step 1 The user registers interest expressions
Step 2 The CM checks new changesets for registered interest expressions
Step 3 If new changesets are available, download and extract changeset files; go to Step 2, otherwise

Listing 1.1: Interest expression example

```
<http://eis.iai.uni-bonn.de/irap/resource/Soccer> rdf:type :Interest ;
        :sourceEndpoint            "http://live.dbpedia.org/sparql" ;
        :lastPublishedFilename     "lastPublishedFile.txt" ;
        :bgp                       "?a  a dbo:Athlete . ?a dbp:goals  ?goals ." ;
        :ogp                       "?a foaf:homepage  ?page ." ;
        :changesetBaseURI          "http://live.dbpedia.org/changesets/" ;
        :changesetPublicationType  "REMOTE" ;
        :subscriber                <http://eis.iai.uni-bonn.de/irap/resource/Sport.org> .
<http://eis.iai.uni-bonn.de/irap/resource/Sport.org> rdf:type :Subscriber;
        :piStoreBaseURI    "sports-pi-tdb" ;    :piStorageType     "TDB" ;
        :targetType        "TDB" ;      :targetEndpoint    "sports-tdb" ;
        :piTrackingMethod  "LOCAL" ;    :targetUpdateURI   "sports-tdb" .
```

Step 4 IE evaluates interest expressions over newly downloaded changeset files
Step 5 IE propagates interesting triples to target dataset
Step 6 IE stores potentially interesting triples to Potentially Interesting Dataset
Step 7 Go to step 2

Interest expression A user registers her interest(s) to Interest Manager (IM) using interest expression ontology[3]. The main part of the expression is the interest query that contains the triple patterns that must be matched to consider a set of triples as interesting. Interest expressions must be defined based on the user's application requirement. That is, interest query of an interest expression must be the same as the query that is used to prepare the initial target dataset (subset of the source dataset).

Changeset management: Once the source dataset URI is registered during interest registration, the Changeset Manager component of iRap regularly checks for new changesets published from the source dataset. Whenever new changesets are found, they are due for download and initiate interest evaluation.

Candidate generation: The first step of interest evaluation over a changeset is generation of candidate interesting triples. These candidates can be set of triples that match the overall or part of interest query. First, the full interest query will be evaluated on the set of deleted triples and added triples, respectively, to get initial interesting deleted triples and interesting added triples. Then, iRap extracts the candidate interesting triples (refers to as potentially interesting triples). To extract potentially interesting triples, we first go through formulation and pruning steps of candidate query.

Query formulation: To obtain all the matching candidate triples from the changeset, we combine basic graph pattern (BGP) of interest expressions and create a CONSTRUCT query that will be evaluated on a changeset. Using combinations, we have $2^{|BGP|} - 1$ candidate queries (excluding the full interest query) for each interest expression.

Pruning: This step removes queries that contain non-disjoint patterns and queries that contains only one triple pattern with three variables. OGPs also be excluded, if they do not have a common variable with the BGP of the candidate query.

[3] http://eis.iai.uni-bonn.de/irap/ontology/

Candidate assertion: Up to this step, iRap computes candidate triples that match part of interest expression, here the missing part of these candidates, with respect to interest expression, will be checked from the target dataset. The outcome of this step confirms whether the candidate triples become interesting with the missing triples can be found in target dataset or not.
Propagation: The propagation step combines the generated candidates with triples that matches the missing parts of interest query which are found in target dataset and apply them on target dataset. Propagation of interesting triples is made sequentially, interesting-removed triples then interesting-added triples. Interesting-removed triples are propagated by applying SPARQL DELETE query on target dataset. Triples that matches the missing parts of the interest query during candidate assertion becomes potentially interesting, because they are not actually removed form the source dataset but they just downgraded to potentially interesting because of they are no longer satisfying the full interest query. Candidate removed triples that do not have matching from target dataset during candidate assertion are propagated by applying SPARQL DELETE query on potentially interesting dataset. Interesting-added triples are propagated by applying SPARQL INSERT query on target dataset. Candidate added triples that do not have matching for the missing parts of interest query from the target dataset are propagated by applying SPARQL INSERT query on potentially interest dataset.

3 Conclusion and Future Work

We presented a framework for interest-based RDF update propagation to consistently maintain replicas of large LOD datasets. We have presented the algorithms used for interest evaluation in our framework. Complementary details and a comprehensive formalization and evaluation of the approach can be found in the accompanying paper [1]. The framework can significantly reduce the size of the data updates required to maintain a local replica up-todate, and the frequency of such updates.

References

1. Endris, K.M., Faisal, S., Orlandi, F., Auer, S., Scerri, S.: Interest-based RDF update propagation. In: 14th International Semantic Web Conference - ISWC2015 (2015)
2. Marx, E., Shekarpour, S., Auer, S., Ngomo, A.N.: Large-scale RDF dataset slicing. In: 2013 IEEE Seventh International Conference on Semantic Computing (2013)
3. Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E., Van de Walle, R.: Querying datasets on the web with high availability. In: ISWC 2014. Springer (2014)