Eurostars Project

3DFed – Dynamic Data Distribution and Query Federation

Project Number: E!114681 **Start Date of Project:** 2021/04/01 **Duration:** 36 months

Deliverable 1.1
Requirement Elicitation and Use Case Specifications

Dissemination Level	Public
Due Date of Deliverable	June 30, 30/06/2021
Actual Submission Date	August 31, 31/08/2021
Work Package	WP1, Requirements Elicitation & Conceptual Architecture
Deliverable	D1.1
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	12

Abstract: This report presents the requirement specification for the 3DFed use cases, which will be the basis for the design of the 3DFed architecture. In particular, we elicit the requirements of the three real world / industry use cases of the project, as defined in work package 5. To this end, we collected requirements from the application areas of the use case partners of 3DFed and based on the collected requirements, we defined the exact specification of the project use cases.

3DFed Project by Eurostars.

History

Version	Date	Reason	Revised by
0.1	06/08/2021	Initial Template & Deliverable Structure	Mirko Spasić & Milos Jovanovik
0.2	09/08/2021	OpenLink Use Case	Milos Jovanovik & Mirko Spasić
0.3	13/08/2021	UPB Use Case	Muhammad Saleem
0.4	16/08/2021	elevait Use Case	Jonas Haupt
0.9	26/08/2021	Alignment of Functional Requirements to Work Packages	Muhammad Saleem, Jonas Haupt, Mirko Spasić & Milos Jovanovik
1.0	31/08/2021	Finalizing	Milos Jovanovik & Mirko Spasić

Author List

Organization	Name	Contact Information
OpenLink Software	Mirko Spasić	mspasic@openlinksw.com
OpenLink Software	Milos Jovanovik	mjovanovik@openlinksw.com
elevait GmbH & Co. KG	Jonas Haupt	jonas.haupt@elevait.de
University of Paderborn	Muhammad Saleem	saleem@informatik.uni-leipzig.de

Contents

1 Introduction

There is an increasing number of datasets used within modern applications that are simply too large to fit into a single server. Therefore, distributed solutions are becoming increasingly popular, more widespread and more frequent than single server configurations. Most often, current distributed solutions are designed for central storage or eventually static data distribution can be applied, which can result in poor query performance. The main goal the 3DFed project is to develop generic approaches and concrete algorithms for the automatic redistribution and federated querying to deal with these large amounts of data.

In this deliverable, we aim to collect the requirements of the 3DFed project from users, use case partners and other stakeholders in order to design the 3DFed architecture accordingly. In particular, we elicit the requirements of the three real world / industry use cases of the project, as defined in work package 5. To this end, we collected requirements from the application areas of the use case partners of 3DFed and based on the collected requirements, we defined the exact specification of the project use cases.

All project partners contribute their own use cases, therefore in this deliverable we collected the following requirements:

- UPB created the requirements and defined the exact specification of the Linked Cancer Genome Atlas (Linked TCGA) use case, as defined in work package 5.
- OpenLink elicited the requirements and defined the exact specification of the LinkedGeoData and DBpedia use case.
- Finally, elevait elicited the requirements and defined the exact specification of the document data used by the product Business Process Automation.

The use cases and collected requirements are described in Section 2. In order to be able to technically evaluate the 3DFed components within the use cases, we also present a collection of measurements derived from the collection of use case requirements, in the same section. In Section 3, we present requirement mapping to the work package in charge of each of the 3DFed components. During the development phase we will steer at reaching those thresholds by recalculating these measures.

2 Use Cases Descriptions

2.1 Linked TCGA

Linked Cancer Genome Atlas (Linked TCGA): Linked TCGA is the RDF version of the Cancer Genome Atlas[1]. This knowledge base contains cancer patient data generated by the TCGA pilot project, started in 2005 by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI). Currently, Linked TCGA comprises a total of 20.4 billion triples[2] from 9000 cancer patients and 27 different tumour types. For each cancer patient, Linked TCGA contains expression results for the DNA methylation, Expression Exon, Expression Gene, miRNA, Copy Number Variance, Expression Protein, SNP, and the corresponding clinical data. Storing such a large dataset in a single endpoint is simply not scalable. In this use case we are aiming to show the actual benefit of our proposed solutions when applied to a real practical use case.

[1] http://cancergenome.nih.gov/
[2] http://tcga.deri.ie/

Table 1: Runtimes (in ms) on large data queries with Virtuoso endpoints. The values inside the brackets show the percentage of the actual query results obtained.(**TO** = Time out after 2.5 hour, **RE** = runtime error).

Qr.	FedX (cold)	FedX (warm)	SPLENDID	ANAPSID	FedX+HiBISCuS	SPLENDID+HiBISCuS
L1	TO (7.2 %)	TO (7.2 %)	123735 (2.73 %)	19672 (15.76 %)	TO (7.2 %)	123700 (2.73 %)
L2	35 (0 %)	35 (0 %)	45473 (1.8 %)	TO (0 %)	76 (0 %)	45479 (1.8 %)
L3	27 (0 %)	27 (0 %)	4877696 (100 %)	TO (0 %)	47 (0 %)	4877991 (100 %)
L4	TO (0.08 %)	TO (0.08 %)	7535531 (0 %)	8775598 (0 %)	62595 (48.34 %)	7535200 (0 %)
L5	TO (0 %)	TO (0 %)	RE (0 %)	TO (0 %)	TO (0 %)	RE (0 %)
L6	TO (0 %)	TO (0 %)	RE (0 %)	TO (0 %)	6127090 (0 %)	RE (0 %)
L7	122633 (100 %)	122500 (100 %)	114456 (100 %)	105447 (100 %)	119449 (100 %)	114400 (100 %)
L8	TO (0.01 %)	TO (0.01 %)	TO (0.05 %)	TO (0.05 %)	TO (0.01 %)	TO (0.05 %)

Elicitation Procedure

Our main requirements for this use case came from the evaluation we performed in LargeRDFBench [6]. We compared five open source SPARQL endpoint federation engines – FedX [8], SPLENDID [3], ANAPSID [3], FedX+HiBISCuS [7], SPLENDID+HiBISCuS [7] – on all of the 32 benchmark queries. The most important finding for *large data queries* is that no system can be regarded as performing better because none of the systems can produce complete results for a majority of the queries. This shows that *current implementations* of query planning strategies (i.e., bushy trees in ANAPSID, left-deep trees in FedX, and dynamic programming in SPLENDID) and join techniques (i.e., adaptive group and dependent join in ANAPSID, bind and nested loop in FedX, and bind, hash in SPLENDID) in the selected systems are not mature enough to deal with large data. In addition, we found that a completeness of the results is not guaranteed in federated SPARQL engines. For example, some queries terminated within the timeout limit and returned zero results due to a flaw in the `FILTER` implementation. In particular, FedX and its HiBISCuS extension give zero results for queries L2, L3, and L5 and send a single endpoint request for each of these queries. All of these queries contain a `FILTER` clause. However, we found that FedX and its HiBISCuS extension are able to retrieve results by removing the `FILTER` clause and setting the `LIMIT=1` in these queries. We also noticed that for queries with incomplete results (e.g., L1, L4 and L8), FedX and its HiBISCuS extension send a large number of endpoint requests and quickly get some initial results. Thereafter, the engines stop sending endpoint requests until the timeout limit is reached. This may be due to some memory leak or possible deadlock in the query execution portion of FedX. Both SPLENDID and SPLENDID+HiBISCuS are able to give complete results for 2/8 large data queries, the highest in comparison to other systems. The query L4 is executed by ANAPSID, SPLENDID, SPLENDID+HiBISCuS within the timeout limit with zero results.

Overall, our fine-grained evaluation points to a major drawback: *while current SPARQL query federation systems can deal with simple and complex queries, they are currently not up to the challenge of dealing with real Big Data queries*, i.e., queries that involve processing large intermediate result sets or lead to large result sets.

Based on the LargeRDFBench results on TCGA data, we define the TCGA use case requirements below.

Requirements

ID	Title	Description	Priority
1-1	Federation of Queries	The proposed 3DFed engine should be able to execute federated SPARQL queries over a set of SPARQL endpoints	High
1-2	Resultset completeness and correctness	The proposed 3DFed engine should be able retrieve complete and correct results	High
1-3	Runtime Efficiency	The proposed 3DFed engine should be able to execute large data queries from LargeRDFBench within reasonable amount of time	High
1-4	Incremental results	The engine produce first few results quickly and presents the remaining results once they are slowly available	Medium
1-5	SPARQL 1.1 support	The 3DFed engine should support full SPARQL 1.1 features	High
1-6	Index update	The 3DFed engine should be able to update its index with underlying dynamic data exchange among triplestores	High
1-7	Scalable data distribution	The 3DFed data distribution engine should be able partition a given Big dataset within a reasonable amount of time	High
1-8	Dynamic data exchange	The dynamic data exchange mechanism should leads to better query runtime performance	High

Benchmark Data (Data and Test Queries)

LargeRDFBench [6] is a a billion-triple benchmark for SPARQL query federation which encompasses real data as well as real queries pertaining to real bio-medical use cases based on TCGA data. LargeRDFBench contains 306 patient data from TCGA. The patients distributed evenly across 3 different cancer types, i.e. Cervical (CESC), Lung squamous carcinoma (LUSC) and Cutaneous melanoma (SKCM). The selection of the patients was carried out by consulting domain experts. This data is hosted in three TCGA SPARQL endpoints with all DNA methylation data in the first endpoint, all Expression Exon data in the second endpoint, and the remaining data in the third endpoint. Consequently, we created three different datasets, namely the Linked TCGA-M, Linked TCGA-E, and Linked TCGA-A containing methylation, exon, and all remaining data, respectively.

LargeRDFBench comprises a total of 32 queries for *SPARQL endpoint federation approaches*. These queries are divided into three different types: the 14 simple queries (namely S1-S14) are from FedBench (CD1-CD7 and LS1-LS7). The 10 complex queries (namely C1-C10) and the 8 large data queries (dubbed L1-L8) were created by the authors with the help of domain experts. The large data queries are from TCGA endpoints. These queries were designed to test the federation engines for real large data use cases, particularly in life sciences domain. These queries span over large datasets (such as Linked TCGA-E, Linked TCGA-M) and involve processing large intermediate result sets (usually in hundreds of thousands) or lead to large result sets (minimum 80459) and large number of endpoint requests. Consequently, the query processing time for large data queries exceeds one hour.

Our main focus will be to text the proposed engine with L1-L8 queries from LargeRDFBench and compare the performance of the 3DFed federation engine with state-of-the-art federation engines such as FedX [8], SPLENDID [3], and ANAPSID [1].

2.2 LinkedGeoData and DBpedia

LinkedGeoData [10] and DBpedia [2, 4] are large-scale RDF datasets (Knowledge Graphs) which have different usage patterns. LinkedGeoData, being the RDF version of the OpenStreetMap data [5], contains geospatial data which have a very specific usage pattern. DBpedia, on the other hand, represents an RDF version of Wikipedia, which in turn has a very broad usage.

This use case will focus on measuring the increase in speed of SPARQL query answering for both LinkedGeoData and DBpedia datasets, and their corresponding infrastructure. The use case goals are to show significant improvement of average SPARQL query response times for both datasets, based on separate use case scenarios for each of them. The planned 3DFed architecture will enable hosting these datasets on a cluster of SPARQL endpoint servers, where automatic and dynamic data and query distribution will enable a significant performance increase for end-users.

Elicitation Procedure

LinkedGeoData and DBpedia datasets are very popular and largely used, with their usage statistics showing continuously rising hits per day. With this, their availability and speed of query answering becomes increasingly important. As the host of the canonical DBpedia SPARQL endpoint and the LOD Cloud Cluster cache of Linked Data datasets which includes LinkedGeoData for more than a decade, OpenLink has access to query logs for these SPARQL endpoints that will be analysed to determine the most commonly used queries and query patterns. This way, a common use case scenario regarding these datasets and real users querying them can be learned and mimicking algorithms can be developed.

In order to specify the use case scenario for the LinkedGeoData dataset, by analyzing the query logs from the current deployment, we will identify common usage patterns. Additionally, we will use existing geospatial benchmarks, e.g. GeoBench [9] and existing example queries from the LinkedGeoData project[3]. These approaches will allow us to define a set of SPARQL queries (or query templates) which mimic usual usage patterns of the dataset, and which we can use to benchmark the performance of the original and the 3DFed-enabled deployment of LinkedGeoData dataset.

Similarly, we will analyze the query logs from the current deployment of DBpedia in order to identify common usage patterns and most accessed parts of the dataset. With this, we will develop a set of SPARQL queries which mimic the real-world usage patterns of DBpedia, and use them to benchmark the DBpedia deployment, as well.

Requirements

The previously mentioned insights led to the requirements presented below.

[3] http://linkedgeodata.org/docs/examples/osm-queries.html

ID	Title	Description	Priority
2-1	LGD Facet Count Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
2-2	LGD Instance Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
2-3	LGD Instance Aggregation Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
2-4	LGD Example Queries	Evaluation time for these types of example queries should be interactive	Medium
2-5	DBPedia Typical Queries	Evaluation time for these types of queries should be reasonable	High
2-6	Query Log Analysis	LGD and DBPedia query logs analysis will be performed in order to identify common usage patterns, queries and query patterns	High

Benchmark Data (Data and Test Queries)

This use case will use the data available in the LinkedGeoData and DBpedia datasets. It will use separate sets of SPARQL queries for benchmarking the two datasets, as outlined above.

2.3 Document Data

elevait's product Business Process Automation (BPA) automatically extracts semantic information from various documents, such as orders and invoices, and stores it in a NoSQL database due to a lack of suitable solutions. With the help of 3DFed, not only the daily received orders (approx. 5000), but also the already archived orders of the last 5 years can be utilised to optimise the automation. For this purpose, approx. 4.8 million documents (constantly increasing) are available at the start of the project. Machine learning (ML) solutions require large-scaled data sets to perform on the required level. On top of the high training run time for the ML models, all ML systems are surrounded by a lot query processes, which stack up to a long processing time. The aim of this use case is to test our solutions in a real world every day ML system.

Elicitation Procedure

In general, the requirements for 3DFed came from our product development team for the BPA solution. Here, the product development is execute in agile development teams, thus, machine learning, service development, ETL processes, and web development. Each team has different requirements regarding accessing the data, e.g., just read or also write access, different kind of complexity of queries, aggregations, speed of the queries, etc. The discussions with the team as well the envisioned roadmap are same drivers for the requirements.

In addition, the current data is not available in a SPARQL endpoint yet. Thus, a general requirement for the use case is the intelligent conversion into RDF. We expect to have at least 250 triplets and about 2 MB of data per document in JSON-LD format, including the decomposition of the image files. Using the integrated 3DFed components, all data should be stored in an appropriate RDF memory allow constant updates. Furthermore, the queries against the NoSQL databases have to be reformulated into SPARQL to show the added values of RDF & SPARQL, moreover the findings offer the possibility to optimise the 3DFed components.

Requirements

The previous discussion lead us to the following requirements.

ID	Title	Description	Priority
3-1	Result completeness and correctness	The engine has to provide complete ad correct results.	High
3-2	Fast write queries	The ETL pipeline has more than 800 processing steps with at least 100 steps that write and update data. This needs to be supported in a scalable manner.	High
3-3	Instance Query	Evaluation time for querying single instances, by ID or other attributes, need to be very low.	High
3-4	Instance Aggregation Query	Especially in the web front-ends and a dashboard, aggregation queries are required. High need to be fast.	High
3-5	Text Search	It needs to be possible to search the content of literals, especially text.	High
3-6	Query Log Analysis	In order to understand the bottlenecks in querying and writing data, it needs to be possible to analyse the query logs.	Medium
3-7	Storage of binary data	Since the data is always extracted from documents, the full-text needs to be stored. Often, the full text is extracted from PDFs or image data. These kinds of data need to be stored and retrieved.	High
3-8	GraphQL API	The web development team is less familiar with the RDF structure and the related SPARQL queries. In order to minimize the learning gap, a GraphQL API would be helpful.	Medium
3-9	Multi-Tenant access	The data of multiple customers should be stored in one scalable database to reduce the deployment and orchestration overhead. Therefore, it is required to logically separate the data securely. A simple tenant (user) management would be beneficial.	Medium
3-10	Scaling	If the data is growing, the data should be partitioned and redistributed in an intelligent manner by collected metrics, like the older data is accessed rarely.	High

Benchmark Data (Data and Test Queries)

As mentioned above, document data is available from different customers where as a large portion is already extracted from the image data. In the beginning, this data is used and could be extended on-demand.

Currently, MongoDB is facilitated as NoSQL database. We have extracted and documented exemplary, heterogeneous queries by the different teams.

3 Alignment of Functional Requirements to Work Packages

Section 2 presented the use case specific requirements. Some of them need to be fulfilled within different 3DFed components and others are use case specific. In the following table, we map the requirements to the corresponding work package.

Task ID	Description	Use Case Requirement ID
WP2 - Data Storage Monitoring and Profiling		
T2.1	Profiles Generation	2-6, 3-2 to 3-4
T2.2	Monitoring the Data Storage Solutions	2-6, 3-2 to 3-4, 3-6
WP3 - Automatic Data Distribution and Dynamic Exchange		
T3.1	Automatic Data Distribution	1-7, 2-1 to 2-5, 3-7 to 3-10
T3.2	Dynamic Data Exchange	1-8, 2-1 to 2-5, 3-7 to 3-10
WP4 - Distributed Query Processing and Optimization		
T4.1	Join-Aware Source Selection	1-3, 1-6, 2-1 to 2-5, 3-7 to 3-10
T4.2	Optimized Query Plan Generation	1-1, 1-2, 1-3, 2-1 to 2-5, 3-6 to 3-10
T4.3	Join Implementation, Pipelining, and Parallelism	1-1, 1-2, 1-3, 1-5, 2-1 to 2-5, 3-6 to 3-10
WP5 - Use Cases		
T5.1	Linked TCGA Use Case	1-1 to 1-8
T5.2	LinkedGeoData and DBpedia Use Case	2-1 to 2-6
T5.3	Document Data Use Case	3-1 to 3-10

4 Conclusion

In this deliverable we discussed the use case specifications pertaining to the 3DFed project. In general, we briefly discussed 3DFed's use cases along with elicitation procedure, requirements, and benchmark data. The summary of this deliverable is provided in the table given below.

Use Case	**Linked TCGA**	**LinkedGeoData & DBpedia**	**Business Process Automation (BPA)**
Description	• Linked TCGA is the RDF version of the Cancer Genome Atlas (TCGA) data. Currently it has over 20 billion triples. • Querying such massive amount of data within reasonable amount of time is challenging. • The goal is to efficiently distribute this data among multiple data nodes and design an optimized query processing engine to efficiently query this data.	• LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF Knowledge Graph • DBpedia dataset contains structured content from the information created in the Wikipedia project and publishes it as an RDF Linked Data Knowledge Graph • Both datasets contain more than a billion triples, which can be challenging hosting on a single server SPARQL endpoint	• Elevait's product Business Process Automation (BPA) works with all kind of documents ad allows for an automated extraction of semantic information. • The data saved in NoSQL databases as JSON. The goal is to make use of the RDF specification for data storage and provide fast read and write operations at scale.
Data Specification	• LargeRDF Bench will be used • Customized TCGA benchmark can be created later on as well	• We will use the existing data from the LinkedGeoData and DBpedia datasets • We will identify existing or develop new sets of SPARQL queries which mimic typical use-case scenarios for using both LinkedGeoData and DBpedia datasets, in order to create a benchmark to test the performance improvement from the 3DFed architecture	• Extracted semantic information of diverse documents (order, invoices, forms, ...) • At least 1 million documents are available.
Mapping Interface	• RDF/XML • Turtle	• RDF/XML • Turtle • XML	• JSON(-LD) • XML

3DFed-Related Metrics	• Resultset completeness and correctness • Query runtime • Network traffic	• Improvement in average query execution times in SPARQL • It will be based on common use-case scenarios for both datasets	• Time/Query • Daily Amount of Queries • Elementary data size • Expected Recall
Expected Results	• Able to execute massive resultset queries within reason amount of time • Efficient data distribution and dynamic exchange mechanisms	• Significant performance improvement of SPARQL queries when accessing the data • Will be measured on both LinkedGeoData and DBpedia datasets	• Showcase usage of scalable handling of RDF data stores in AI-driven enterprise software stack • Similar or better performance in the ETL pipeline • Improved performance on querying linked data artifacts
Expected Impact of 3DFed	Advances state of the art regarding: • Data distribution • Dynamic data exchange • Efficient SPARQL query processing	• Automatic and dynamic data distribution across a cluster of SPARQL endpoint servers • Faster data access for the largely popular LinkedGeoData and DBpedia datasets	• RDF-based storage allows to bring up new use case for any customers • Lower query time $\rightarrow$ Faster processing of big data

References

[1] Maribel Acosta, Maria-Esther Vidal, Tomas Lampo, Julio Castillo, and Edna Ruckhaus. ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 18–34. Springer Berlin Heidelberg, 2011.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] Olaf Görlitz and Steffen Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VoID Descriptions. In *O. Hartig, A. Harth, and J. F. Sequeda, editors, 2nd International Workshop on Consuming Linked Data (COLD 2011) in CEUR Workshop Proceedings*, volume 782, October 2011.

[4] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6, 01 2014.

[5] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, accessed on: 11.08.2021.

[6] Muhammad Saleem, Ali Hasnain, and Axel-Cyrille Ngonga Ngomo. Largerdfbench: a billion triples benchmark for sparql endpoint federation. *Journal of Web Semantics*, 48:85–125, 2018.

[7] Muhammad Saleem and Axel-Cyrille Ngonga Ngomo. HiBISCuS: Hypergraph-Based Source Selection for SPARQL Endpoint Federation. In Valentina Presutti, Claudia d'Amato, Fabien Gandon, Mathieu d'Aquin, Steffen Staab, and Anna Tordai, editors, *The Semantic Web: Trends and Challenges*, volume 8465 of *Lecture Notes in Computer Science*, pages 176–191. Springer International Publishing, 2014.

[8] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 601–616. Springer Berlin Heidelberg, 2011.

[9] Mirko Spasić. Design of Geospatial Benchmarking System and Performance Evaluation of Virtuoso and PostGIS. In Milan Zdravković, Miroslav Trajanović, and Zora Konjović, editors, *Proceedings of ICIST 2015 - 5th International Conference on Information Society and Technology*, volume 1, pages 154–159. Society for Information Systems and Computer Networks, 2015.

[10] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*, 3:333–354, 01 2012.