Big data batch processing
M2 LID and SIA
2022-23

DataFrame and Spark SQL over LUBM1

1

We are processing an RDF (Resource Description Framework) data set from the LUBM benchmark with a parameter set at one university. The dataset (univ1.nt) has more than 100,000 triples which are made up of URIs and literals. Each tripet is made up of a subject, a property and an object. These 3 components are separated by a space (' '). We also provide two dictionaries: one for concepts (univConcepts.txt) and one for properties (univProps.txt). Entries in these dictionaries correspond to a URI and an integer value (the URI identifier). The principle of the exercise is to encode the dataset so as to replace the URIs of the properties by an identifier (from the dictionary of properties) and the same for the concepts. Concepts can only appear at the object position of a triplet (especially when the property is 'type'). For example, the following 2 triples:

```
<http://www.Department12.University0.edu/GraduateStudent9>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
        <http://www.univ-mlv.fr/~ocure/lubm.owl#ResearchAssistant> .
<http://www.Department12.University0.edu/GraduateStudent9>
        <http://www.univ-mlv.fr/~ocure/lubm.owl#advisor>
        <http://www.Department12.University0.edu/FullProfessor6> .
```

since the `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type` and `<http://www.univ-mlv.fr/~ocure/lubm.owl#advisor>` properties are respectively mapped to 0 and 1233125376; and that `<http://www.univ-mlv.fr/~ocure/lubm.owl#ResearchAssistant>` is mapped to 1015021568, the 2 triples should be transformed in the following triples:
1 0 1015021568
1 1233125376 2
the 1 and 2 identifiers are set arbitrary bu the system.

In this first part, we will only handle RDDs. Whenever you create an RDD, consider whether it is worth persisting it. Also consider the stages where an unpersist might be relevant. Check the Spark storage states on the web interface: `http://localhost:4040/storage`

1.1

Load the dataset and the two dictionaries. Provide the sizes of these 3 RDDs.

1.2

Encode the properties of the dataset. Provide the number of properties used in this new version of the dataset. Are all dictionary properties used in the dataset?

1.3

You need to create the dictionary of individuals (that is, the dataset entries that are not properties or concepts). Each individual must have a unique identifier. To assign these identifiers, you can use one of the RDD zip methods (on http://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.rdd.RDD). How many unique individual do you have in this RDD.

1.4

You have sufficient dictionaries (concepts, properties and individuals) to now encode the rest of the dataset. Starting from the RDD created in 1.2, transform the URIs and individuals into their respective identifiers.

1.5

From the RDD of 1.4, encode the subjects of the triples. Check that no triplets were lost during processing by providing the number of entries in this last RDD. Check the size of the encoded dataset. Was it worth encoding this data set in terms of memory footprint?

2

We will now take advantage of this data set to execute queries (in SQL then on the DSL of DF). We consider the following query (Q1) (expressed in SPARQL, the W3C language associated with the RDF data model):

```
SELECT ?x ?y ?z
WHERE {?x lubm:advisor ?y.
       ?y lubm:worksFor ?z.
       ?z lubm:subOrganisation ?t.}
```

The way SPARQL works is based on "graph pattern matching". In fact, our lubm1 dataset is a graph, and the WHERE clause also describes a graph. We therefore look for assignments for the variables (symbols prefixed with a '?') by looking for query graph pattern in the dataset. The SELECT clause indicates (as in SQL), the variables that we want to see in the result. The WHERE clause describes a conjunction of triples where each triple follows the pattern "subject property object.". The property of the first triplet of Q1 is therefore 'advisor' (whose identifier in our dictionary is: 1233125376. The encoded version of Q1 becomes (now denoted Q1') According to our dictionaries, the encoded version of SPARQL is:

```
SELECT ?x ?y ?z
WHERE {?x 1233125376 ?y. ?y 1136656384 ?z. ?z 1224736768 ?t.}
```

2.1

From the encoded LUBM1 encoded triple of Section1, create a DataFrame with a schema corresponding to "sub", "pred" and "obj". Create a table that will be used to execute SQL queries over. What is the impact in terms memory footprint being used to transform into a Dataframe?

2.2

Translate query Q1' in SQL. Execute it. How many results do you get?

2.3

The result set is encoded, decode it using the dictionaries.

2.4

Translate the query into the DataFrame DSL

2.5

Provide the property distribution in the LUBM1 dataset. This corresponds to the number of occurrences of each property in the dataset. What is the most frequently used property, what is the occurrence number?

2.6

Display a graphical representation for this property distribution