STATisfy Me: What are my Stats?

Gezim Sejdiu[1], Ivan Ermilov[2], Mohamed Nadjib Mami[1,3] and Jens Lehmann[1,3]

[1] Smart Data Analytics, University of Bonn, Germany
sejdiu@cs.uni-bonn.de, mami@cs.uni-bonn.de, jens.lehmann@cs.uni-bonn.de
[2] Department of Computer Science, University of Leipzig, 04109 Leipzig, Germany
iermilov@informatik.uni-leipzig.de
[3] Fraunhofer IAIS, Germany
mohamed.nadjib.mami@iais.fraunhofer.de, jens.lehmann@iais.fraunhofer.de

Abstract. The increasing adoption of the Linked Data format, RDF, over the last two decades has brought new opportunities. It has also raised new challenges though, especially when it comes to managing and processing large amounts of RDF data. In particular, assessing the internal structure of a data set is important, since it enables users to understand the data better. One prominent way of assessment is computing statistics about the instances and schema of a data set. However, computing statistics of large RDF data is computationally expensive. To overcome this challenging situation, we previously built DistLODStats, a framework for parallel calculation of 32 statistical criteria over large RDF datasets, based on Apache Spark. Running DistLODStats is, thus, done via submitting jobs to a Spark cluster. Often times, this process is done manually, either by connecting to the cluster machine or via a dedicated resource manager. This approach is inconvenient as it requires acquiring new software skills as well as the direct interaction of users with the cluster. In order to make the use of DistLODStats easier, we propose in this paper an approach for triggering RDF statistics calculation remotely simply using HTTP requests. DistLODStats is built as a plugin into the larger SANSA Framework and makes use of Apache Livy, a novel lightweight solution for interacting with Spark cluster via a REST Interface.

1 Introduction

SANSA [3] is an open source framework[4] that allows RDF processing at scale. It provides a set of libraries for executing SPARQL queries, performing inference as well as analytics over knowledge graphs, all while supporting several RDF representations. In addition, it provides support for RDF dataset statistics and quality assessment for large-scale RDF datasets. The statistics are calculatated using the dedicated component DistLODStats [4], which is a distributed and scalable software able to compute 32 statistical criteria (intially proposed at [1]).

SANSA and DistLODStats use Apache Spark[5] as an underlying engine, which is a popular framework for processing large datasets in-memory. Spark provides two possibilities of running and interacting with applications:

[4] https://github.com/SANSA-Stack
[5] http://spark.apache.org/

– *Interactive* - via a command line interface (CLI) called *Spark Shell*, or via Spark *Notebooks* (e.g. SANSA-Notebooks [2]),
– *Batch* - which includes a bash script called *spark-submit* used to submit a Spark application to the cluster without interaction during run time.

Spark application is usually launched by logging first into a cluster, either in the premises or remotely in the cloud. This process presents several difficulties:

– It requires a sophisticated user access control management, which may become hard to maintain with multiple users.
– It raises the chances of exhausting the cluster or even causing its failure.
– It exposes cluster and its configurations to all the users with access.

In order to elevate those, we have investigated Apache Livy [6] – a novel open source REST interface for interacting remotely with Apache Spark. It supports executing snippets of code or programs in a Spark context that runs locally, in a Spark cluster or in Apache Hadoop YARN.

This is an accompanying poster paper for DistLODStats [4], which was accepted at the ISWC resource track. The addition made in this poster is an interactive REST API for DistLODStats, which enables calculating RDF dataset statistics remotely i.e., without a direct contact with the hosting cluster.

2 STATisfy: A REST Interface for DistLODStats

Traditionally, when running a Spark job, submitting it to a Spark cluster is done via a *spark-shell* or *spark-submit*. Usually, this process is done manually either entering the cluster gateway machines or via a dedicated resource manager (e.g. SLURM, OpenStack).

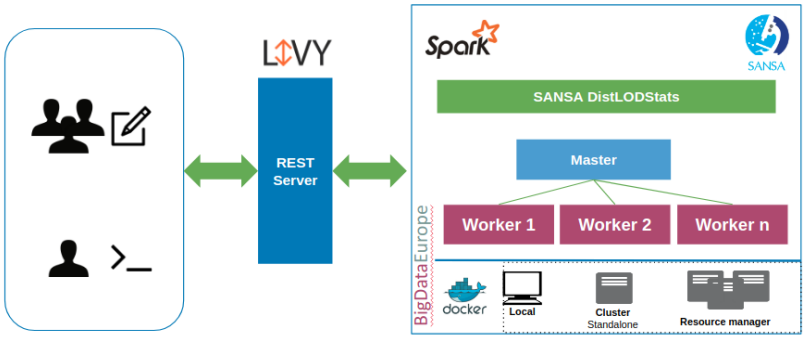

Fig. 1. STATisfy overview architecture.

[6] https://livy.incubator.apache.org/

For users with little experience in cluster management and the Hadoop infrastructure, it can be challenging to run Spark. As an alternative, we introduce **STATisfy**[7]: REST Interface for DistLODStats. Instead of computing RDF statistics directly on the cluster the interaction is done via REST APIs (as it is depicted in the Figure 1).

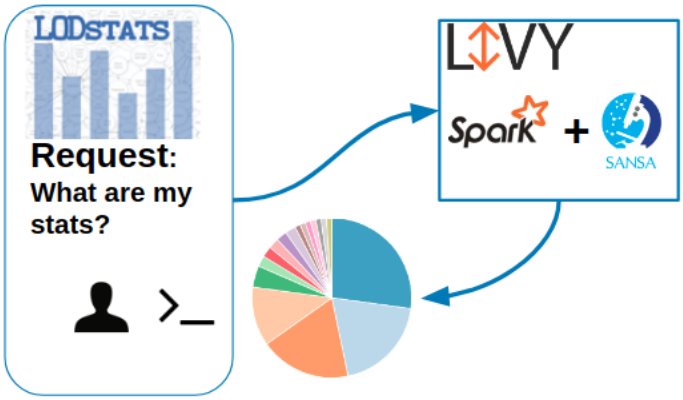

Fig. 2. STATisfy used on LODStats system.

The client side will create a remote Spark cluster for initialization, and submit jobs through REST APIs. Livy REST Server will then discover this job and send through remote procedure call (RPC) to SparkSession, where the code will be initialized and executed. In the meantime, the client will be waiting for the result of this job coming from the same direction.

Running the STATisfy is similar to using DistLODStats via *spark-submit*. The difference is that this shell is not running locally, instead, it runs in a cluster and transfers the data back and forth through the network.

For demonstrating the usage of the tool, we have deployed it on the comprehensive statistics catalogue LODStats[8] which crawls RDF data from metadata portals such as CKAN dataset metadata registry. By doing this, it obtains a comprehensive picture of the current state of the Web of Data. As we use DistLODStats as an underlying engine for computing RDF statistics afterwards, the limitation was that the user has to interact with the cluster manually and initiate the job for computing such statistics. By using STATisfy REST interface, LODStats will interact with the cluster from anywhere which provides the capabilities necessary to do this without compromising on ease of use or security.

As it is shown on the Figure 2, user starts a session via REST API using Livy for submitting a job to the Spark cluster.

[7] https://github.com/GezimSejdiu/STATisfy
[8] http://lodstats.aksw.org/

Listing 1.1. DistLODStats example REST call.

```
curl -X POST -H "Content-Type:application/json"
sansa-stack.net:8998/batches --data '{
      "file": "hdfs:.///tmp/REST/sansa-rdf-stats.jar",
      "className": "net.sansa_stack.examples.spark.rdf.RDFStats",
      "name": "SANSA RDF Dataset Statistics",
      "executorCores":1, "executorMemory":"512m", "driverCores":1,
      "driverMemory":"512m", "args":["-i hdfs:.///input.nt"]}'
```

The script (see Listing 1.1) contains a spark-submit configurations which is given in the format of a JSON structure with the necessary information like spark-submit. With the POST request ***POST /batches*** user could submit a request to DistLODStats using Livy server. Using Livy, STATisfy will then help to launch this request in the cluster. As a result, the output will be curled by their end in the format of VoID description.

3 Conclusions

In order to deepen their understanding of the data, many users require gathering statistical information about RDF datasets. This process becomes compute-intensive when the datasets grow in size. DistLODStats is a prominent solution, however, it requires setup and managing of the the cluster configuration and job submission. To make the process easier, we have introduced STATisfy, a tool for interacting with DistLODStats via a REST Interface. This way DistLODStats can be provided as-a-service, where users only send (HTTP) requests to the remote cluster and obtain the wished results, without having any knowledge about system access or cluster management. STATisfy is used for the LODStats project and an inclusion in the new DBpedia[9] community release processes is ongoing.

References

1. J. Demter, S. Auer, M. Martin, and J. Lehmann. Lodstats—an extensible framework for high-performance dataset analytics. In *Proceedings of the EKAW 2012*, Lecture Notes in Computer Science (LNCS) 7603. Springer, 2012.
2. I. Ermilov, J. Lehmann, G. Sejdiu, L. Bühmann, P. Westphal, C. Stadler, S. Bin, N. Chakraborty, H. Petzka, M. Saleem, A.-C. N. Ngonga, and H. Jabeen. The Tale of Sansa Spark. In *Proceedings of 16th International Semantic Web Conference, Poster & Demos*, 2017.
3. J. Lehmann, G. Sejdiu, L. Bühmann, P. Westphal, C. Stadler, I. Ermilov, S. Bin, N. Chakraborty, M. Saleem, A.-C. N. Ngonga, and H. Jabeen. Distributed Semantic Analytics using the SANSA Stack. In *Proceedings of 16th International Semantic Web Conference*, 2017.
4. G. Sejdiu, I. Ermilov, J. Lehmann, and M. Nadjib-Mami. DistLODStats: Distributed Computation of RDF Dataset Statistics. In *Proceedings of 17th International Semantic Web Conference*, 2018.

[9] https://wiki.dbpedia.org/