Sophia: a Linked Data and Semantic Web toolkit for Rust

Pierre-Antoine Champin

Sophia: a Linked Data and Semantic Web toolkit for Rust

Author: Pierre-Antoine Champin (LIRIS – Université de Lyon)

This presentation is about Sophia[1], an open-source implementation in the Rust[2] programming language of Linked Data and Semantic Web technologies. It aims to provide Rust developers with basic tools and extensible APIs to take full advantage of these technologies. At the same times, it relies on Rust's strengths to provide an efficient and reliable toolkit for Linked Data and Semantic Web practitioners.

Why Rust?

Rust is a relatively recent programming language (the first stable compiler was released in 2012), sponsored by Mozilla, and emphasizing performance, reliability and productivity. It is a statically typed compiled language, providing high-level abstractions (such as iterators and generic types) while preserving low-level control and execution speed. Its unique way of managing memory at compile time (thanks to the *borrow checker*) protects programmers against a range of nasty bugs, such as dangling pointer dereferencing or race conditions in concurrent programming. Despite a steep learning curve, Rust has been elected "most lovable programming language" in the Stack Overflow Developer Survey every year since 2016[3].

Sophia is efficient

Although it is still at an early stage, Sophia already has very competitive performances, compared to mature implementations such as Jena[4], librdf[5], N3.js[6] or RDFlib[7]. Figures 1, 2 and 3 show that Sophia is faster than these implementations, both for loading an RDF graph in memory and for answering simple queries over this graph. They also show that Sophia is the only one among these libraries able to load up to 10^7 triples in memory. Finally, Sophia has the smallest memory footprint (on par with librdf). These very encouraging first results must be credited to the optimization abilities of the Rust

[1] https://github.com/pchampin/sophia_rs
[2] https://www.rust-lang.org/
[3] 2016: https://stackoverflow.com/insights/survey/2016#technology-most-loved-dreaded-and-wanted
2017: https://stackoverflow.com/insights/survey/2017#most-loved-dreaded-and-wanted
2018: https://insights.stackoverflow.com/survey/2018#most-loved-dreaded-and-wanted
2019: https://insights.stackoverflow.com/survey/2019#most-loved-dreaded-and-wanted
[4] Implemented in Java, http://jena.apache.org/
[5] Implemented in C, http://librdf.org/
[6] Implemented in Javascript, https://github.com/rdfjs/N3.js
[7] Implemented in Python, rdflib.readthedocs.org/

compiler (itself based on the LLVM[8] infrastructure), as well as to the high quality of the Rust standard library. These results, with the code to reproduce them, are available at https://github.com/pchampin/sophia_benchmark/.

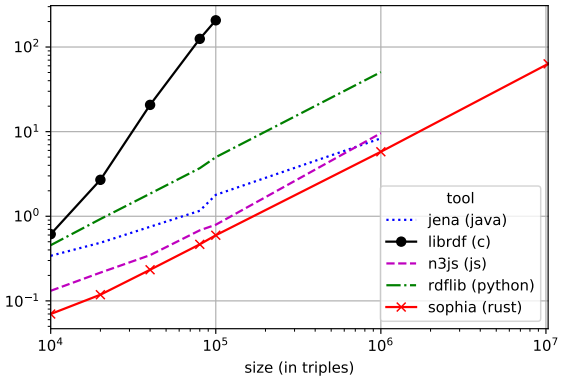

Figure 1: Time (in s) to load an NT file in memory

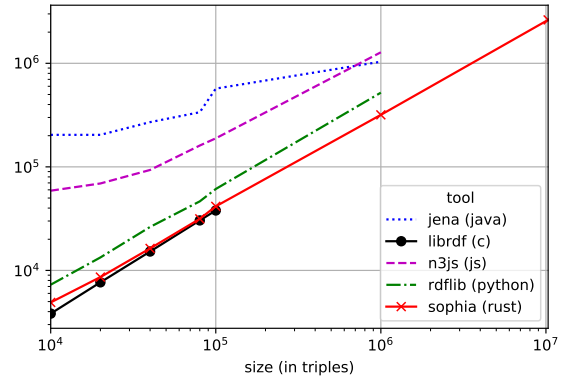

Figure 2: Memory (in kB) used to load an NT file in memory

[8] https://llvm.org/

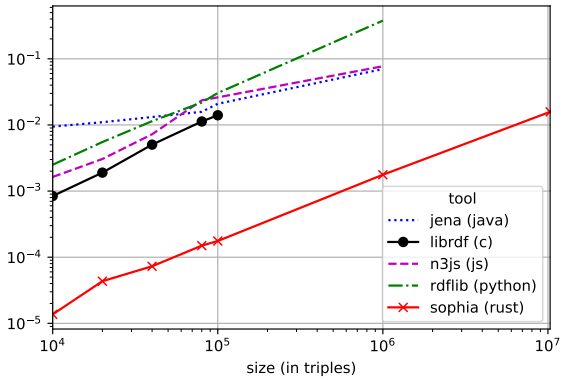

Figure 3: Time (in s) to retrieve all triples matching (*,p,o)

Sophia is portable

The Rust compiler can target a number of hardware/software architectures, and has several options for working with lightweight devices (e.g. not linking the standard library, or using a custom memory allocator). This makes Rust a good programming language for the Web of Things, and Sophia could prove valuable in this context.

Furthermore, Rust is the language of choice for compiling to Web Assembly[9]. A suite of tools, maintained by a very active community, makes it very easy to compile and deploy Web Assembly modules written in Rust and able to interoperate with Javascript code. One of our goal is to use Sophia for building efficient in-browser RDF applications.

Perspectives

Currently, Sophia is able to parse a few RDF syntaxes (including RDF/XML, Turtle and TriG) and to serialize to N-Triples. Triples are stored in memory, and can be queried against simple triple patterns. In the future, we want to support more formats (in particular JSON-LD), persitent triple storages, and other RDF-related standards such as SPARQL, RDFS, OWL, and SHACL.

There is obviously a long way to go, but this can be achieved as a community effort. First, as mentioned above, Sophia is an open-source project, which has already raised interest from a few contributors. Second, Sophia's goal is not to become a big monolithic project implementing all of these technologies at once. It aims to lay the foundations of a diverse ecosystem of interoperable components. It defines a number of *generic and reusable* interfaces (*traits* in Rust parlance) for all the basic concepts underlying RDF (triple, graph, dateset, etc.), that others can implement. In that sense, it follows the philosophy of the RDF-JS initiative[10]. We have already started to interact with another RDF-related Rust project[11] in order to determine how to best collaborate and complement each other.

[9] https://webassembly.org/
[10] https://rdf.js.org/
[11] https://github.com/Tpt/rio