Smart Data Filtering in the RIPE Atlas Measurement Platform

Spyridon-Andreas Siskos
asiskos@ics.forth.gr
University Of Crete / ICS-FORTH

Petros Gigis
p.gkigkis@cs.ucl.ac.uk
University College London / ICS-FORTH

Lefteris Manassakis
leftman@ics.forth.gr
ICS-FORTH

Xenofontas Dimitropoulos
fontas@ics.forth.gr
University Of Crete / ICS-FORTH

1 INTRODUCTION

Researchers and network operators use data plane measurement tools (e.g. traceroute, ping) for network monitoring and diagnostics. Moreover, platforms such as RIPE Atlas [2] and CAIDA's Archipelago [1] provide access to data plane measurement tools by leveraging the highly distributed footprint of their Vantage Points (VPs) around the globe.

We focus on extending the functionality of the RIPE Atlas platform, since it has the largest amount of VPs, reaching ~11k worldwide [2], allows users to make their own measurement campaigns, and also provides access to the results through a structured REST API.

The structure of the RIPE Atlas API allows users to collect the results of their own measurements. However, it does not provide advanced filtering that would allow users to easily pinpoint and explore results generated by other users. The current API supports filtering per target address and a limited number of parameters (e.g. start/end timestamp). It lacks more advanced and sophisticated filtering. For example, users are interested in results between specific source-destination Autonomous Systems (AS) pairs. By getting the data filtered only by destination as the current practice dictates, it adds significant overhead to users, since they must download and parse a vast amount of results that are irrelevant to their use case.

In this work, we aim to provide an open-source framework on top of the RIPE Atlas API that will enable advanced filtering of results and will allow users to explore the RIPE Atlas collected data more efficiently.

2 FRAMEWORK WORKFLOW

The current prototype is developed using Python3, PostgreSQL and the Django Framework combined with the Swagger API as multi-container Docker application. As shown in Fig.1 we download and crawl the measurement metadata from the RIPE Atlas platform. We analyze and enrich them by adding extra meta-information that will later allow us to perform advanced queries. Specifically, we add current VP information (e.g. AS, IP prefix, location). We finally store the enriched data into a relational database which is accessible through a structured REST API.

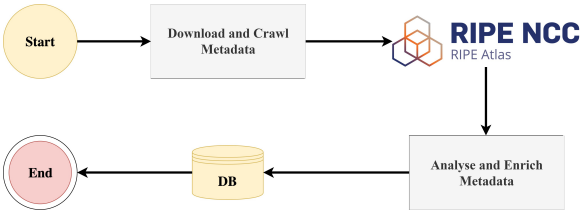

Figure 1. Framework Workflow

Fig.2 depicts the user-prototype interaction.

1. User issues an advanced query to our API (e.g. fetch all traceroutes from AS8522 towards AS32934).
2. The tool translates the query to a set of URLs containing appropriate filter parameters pointing to the RIPE Atlas API, which will return only the related results.
3. By iterating to the set of these URLs, user retrieves all available results matching his initial query.

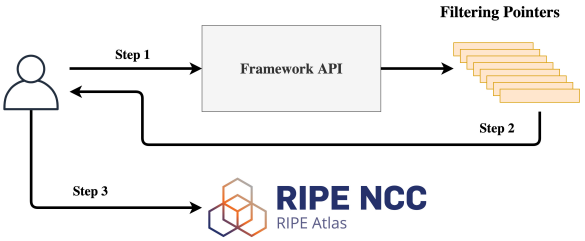

Figure 2. Workflow of a user request

3 CONCLUSIONS AND FUTURE WORK

Although RIPE Atlas is a well-established measurement platform, there is still room for improvement in result filtering. As future work, we plan to extend our API with more measurement types, improve performance and scalability, and use the RIPE Atlas Stream [3] for real time result processing.

References

[1] CAIDA. 2020. Archipelago (Ark) Measurement Infrastructure. https://www.caida.org/projects/ark/.
[2] RIPE NCC. 2020. RIPE Atlas Platform. https://atlas.ripe.net.
[3] RIPE NCC. 2020. RIPE Atlas Result Streams. https://atlas.ripe.net/docs/result-streaming/.