Analyzing and Comparing DNS Lookup Tools in Python

José C. C. Pinto, Eder J. Scheid, Muriel F. Franco, Lisandro Z. Granville

Informatics Institute (INF) – Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brazil

{jccpinto,ejscheid,mffranco,granville}@inf.ufrgs.br

Abstract. *The performance of Domain Name System (DNS) resolvers is crucial, as the majority of the communication in the Internet starts with a DNS lookup to resolve a domain an IP address to reach the desired content. In this sense, the academia has been devoted to measure and analyze the performance of DNS resolvers using different tools. However, such tools might present different results due to their implementation and affect the measurements. Hence, this paper provides an analysis and comparison of there different DNS lookup tools employed in the literature and discuss the impact of the tool selection.*

1. Introduction

Established in 1983, the Domain Name System (DNS) emerged as a critical component of the Internet [Mockapetris and Dunlap 1988]. Its primary function is to translate user-friendly hostnames (*e.g.*, *google.com*) into their corresponding Internet Protocol (IP) addresses, effectively serving as the "phone book" of the Internet [Kurose and Ross 2016]. Nearly all Internet communication starts with a DNS lookup, and complex websites which require content from multiple third parties might perform hundreds of DNS requests before loading a single page [Butkiewicz et al. 2011]. Thus, DNS performance is of concern as it directly impacts performance in most Internet-based communications [Bozkurt et al. 2017].

Past work has measured DNS performance extensively and under different conditions. For example, [Ager et al. 2010] thoroughly analyzed the performance of DNS with distributed measurements across more than 50 different ISPs, in over 28 countries, comparing local and public DNS resolvers. [Böttger et al. 2019] focused on comparing performance between DNS and its encrypted versions, DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH), and their impact in webpage loading times. [Affinito et al. 2022] measured DoH performance overhead as well as malicious domain protection. However, they all use different DNS lookup tools (*e.g.*, dig, dnspython, and pydig). When evaluating DNS performance, the lookup tool used might introduce additional overhead and skew results, underscoring the necessity for careful tool selection when designing experiments.

In this paper, we compare the performance of different DNS lookup tool libraries in Python. For that, we review the literature to gather the most common used tools and select three of them as focus of our analysis. We collect a dataset of measurements by performing several DNS queries to different public resolvers using the selected tools. Our analysis focuses on Response Time (RT), which is the time elapsed between issuing the DNS request and receiving a response. Our objective is to determine the impact that tool selection could have on DNS performance measurements.

The rest of this paper is structured as follows. Section 2 resents an overview of the DNS and its design. Section 3 describes the tool selection for the experiments. Section 4 details the methodology employed in the measurements and its implementation. Section 5 presents the experiment setup and discusses the results obtained. Lastly, Section 6 summarizes key findings and suggests future work.

2. Domain Name System (DNS)

The DNS is comprised of two main components: a distributed database, structured as a hierarchy of DNS servers, and an application-layer protocol that allows end-hosts (*i.e.*, clients) to query the database, by using local name servers called Resolvers [Kurose and Ross 2016]. To reach a global scale, the DNS database is comprised by a large number of servers distributed around the world, and no single DNS server has the information of all hosts in the Internet. In this sense, the structure of the database is similar to that of the Unix file system, as an inverted tree with the root at the top, and is indexed by domain names. The information, such as IP address, associated with each domain name is stored in Resource Records (RR) [Liu and Albitz 2001].

When a software requires information from the domain namespace, such as a Web browser that needs to translate the domain *www.inf.ufrgs.br* to its IP address, it invokes the client side of DNS resolver, initiating the queries that compose the so-called resolution process. Because of the inverted tree structure of the domain namespace, any domain can be reached by starting the search at the root nameservers. The resolver queries the root servers, which queries the Top-level Domain (TLD) servers (*e.g.*, `.br`) and down the name space tree of servers until an authorative server for the organization (*e.g.*, `.inf`) can return the IP address for *www.inf.ufrgs.br*. Thus, completing the resolution process.

Without additional information, queries start at the root name servers, making them essential to the DNS. However, to offload some of that heavy traffic, caching is very important, as it prevents name servers from querying root nameservers each time it receives a request for an answer it does not have locally. Additionally, caching increases the speed of name resolution as in any part of a query chain a DNS server might have cached the required answer or the address of the authoritative nameserver for the zone, therefore, reducing the number of required queries for resolution [Liu and Albitz 2001].

3. Selection of Lookup Tools

To select the tools to be analyzed in this work, we reviewed the literature on research approaches that focused on analyzing the performance of DNS resolvers, as we were unable to find work on comparing lookup tools directly. Table 1 presents such a review. While all of the works focused measuring DNS performance, they varied in objective and scope. [Affinito et al. 2022] and [Ager et al. 2010] focused on comparing local and public resolver performance. [Böttger et al. 2019], [Sharma et al. 2022], and [Doan et al. 2021] investigated the performance impact of using encrypted DNS through HTTPS or TLS protocols, while [Hounsel et al. 2020] and [Borgolte et al. 2019] do so with additional attention to Web page loading times.

Regarding DNS lookup tools, we could observe different approaches, including the use of proprietary monitoring software such as SamKnows and BrightData but also

a non-commercial distributed monitoring tool, called RIPE Atlas. For this work, we selected open-source and accessible tools, which are the Python libraries *pydig 0.4.0* and *dnspython 2.4.2*, as well as the native *dig* Linux command. Both are available through *PyPI* and *Github* and present good enough documentation, and were relatively simple to setup and experiment with.

Table 1. Review of Literature on DNS Resolvers Performance Research

Reference	Protocol	Lookup Tool	List of Analyzed Resolvers
[Affinito et al. 2022]	Do53, DoH	pydig	Google, OpenDNS
[Böttger et al. 2019]	Do53, DoH, DoT	dnspython	Several Resolvers
[Ager et al. 2010]	Do53	dig	Google, OpenDNS
[Hounsel et al. 2021]	Do53, DoH, DoT	SamKnows	Anonymized Public Resolvers
[Hounsel et al. 2020]	Do53, DoH, DoT	dns-measurement	Google, Cloudflare, Quad9
[Borgolte et al. 2019]	Do53, DoH	Firefox	Google, Cloudflare, Quad9
[Sharma et al. 2022]	Do53, DoH	dns-measurement	Several Resolvers
[Chhabra et al. 2021]	Do53, DoH	BrightData	Google, Cloudflare, Quad9, NextDNS
[Doan et al. 2021]	Do53, DoT	RIPE Atlas	Google, Cloudflare, Quad9, CleanBrowsing, UncensoredDNS

DNS-over-Port 53 (Do53)

4. Methodology and Implementation

Our resolver dataset consists of 6 widely used public resolvers which we also derived from literature research (*cf.* Table 1), being Google, Cloudflare, Quad9, Cisco, CleanBrowsing, and Adguard. For our domain dataset, we selected the most popular domain of the Tranco list [Le Pochat et al. 2019], retrieved from September 13, 2023, which was *www.google.com*. For each one of the DNS resolvers, the lookup to the selected measurements was performed in a loop 500 times so that a significant sample size and confidence interval could be gathered during analysis.

To issue DNS queries from all the tools selected, we implemented a measurement tool that executes and collects the results of each combination of tools (*e.g.*, dig, pydig, and dnspython), resolvers (*e.g.*, Google, Quad9, and Adguard) and domains (*e.g.*, *google.com* and *ufrgs.br*). The tool stores the results as CSV files for posterior analysis. The performance metric relevant for this study is the Response Time (RT) of a lookup, which consists in the time elapsed between issuing the query and receiving a response from the resolver. To obtain accurate RTs we measure them using Python's `time` module for each of the tools selected.

5. Results and Discussion

The setup of the experiment was an 2.3 GHz Intel® Core™ i5-6200U machine running on a Linux Debian 11 operating system, with 16 GB of RAM. The experiments were performed in a single home connection using an Ethernet cable connected to the Internet. All source code, domain and resolver datasets, as well as measurement results are available at `https://github.com/jchagastelles/encrypted-dns-benchmark`.

The experiment results are depicted in Figure 1. For each resolver in the *x*-axis, different bars are depicted, and the *y*-axis represents the response time, in milliseconds, measured by each tool. The *dig_awk* bar represents the measurements using the *dig* command to get the DNS query time, *dig_timelib* represents the measurements using the Python's `time` library [Python Software Foundation 2023] and calling *dig* from the

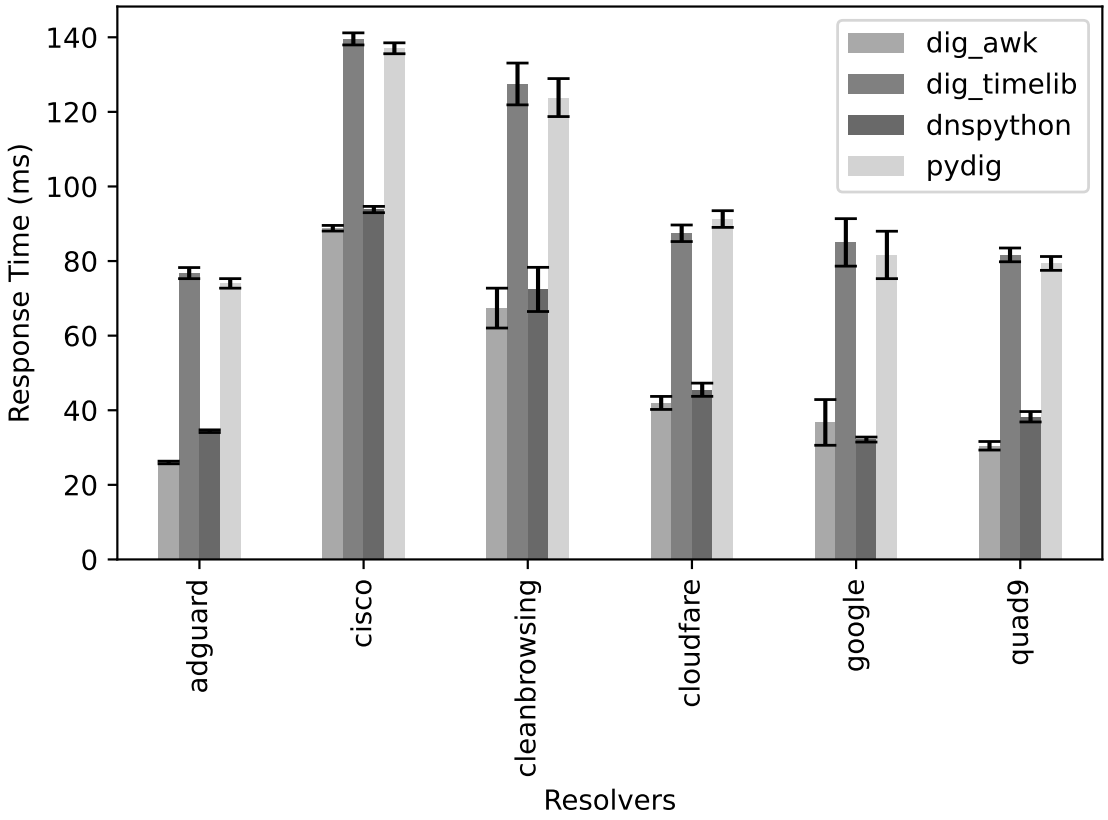

Figure 1. Results of the Experiments using Different DNS Lookup Tools

Python's `subprocess` module [Astrand 2003], the *dnspython* bar represents the measurements performed using the *dnspython* library and, lastly, the *pydig* bar represents the measurements with the *pydig* library.

It can be seen that the measured RTs varied both for different tools and for different resolvers. For the same resolver, we can observe RT differences as high as 153% between different tools, as can be seen in the case of *dnspython* (32ms) and *pydig* (81ms) mean RTs for the Google resolver. On average, we can see that the best performance is from the native *dig*'s reported query RTs, closely followed by *dnspython*. *Pydig* performance was consistently worse, and similar to that of our implemented *dig* tool measurements using the Python time module.

One reason that might explain the performance differences between *dnspython* and *pydig* is the fact that *pydig* acts as a wrapper to *dig*, using the `subprocess` module; thus, it requires system calls (*e.g.*, opening a process and reading from process descriptors) from Python to the Operating System (OS). In contrast, *dnspython* performs the queries directly in native Python code by relying on native UDP and TCP sockets, which results in a faster communication with the resolver compared to *pydig*. Thus, it shows that *dnspython* is the closest one to the native *dig* command (*i.e.*, *dig_awk* in Figure 1).

6. Conclusion and Future Work

In this paper, we analyzed the performance impact of using different DNS lookup tools in DNS performance measurements. The literature on DNS performance measurement

was researched to investigate and select which were the most employed tools and DNS resolvers in the approaches.Based on that, three tools and six DNS resolvers were selected for our analysis.

To perform the experiments in a reproducible manner, we designed and implemented a tool that performs DNS lookups using the selected tools to the six resolvers several times. From the experiments's results, we found significant variation in lookup RTs across different tools and resolvers, with performance impacts as high as 153%.

Based on our findings, we conclude that the tool selection directly impacts results when analyzing DNS performance. This difference in results can be explained due to the tool's implementation, which varies from using the OS to call an external DNS lookup tool (*e.g.*, *dig*) or using native Python sockets to create the DNS requests (*e.g.*, *dnspython*). Thus, it is suggested that researchers carefully select the tool when designing future experiments and take into consideration that results might be impacted.

Future work on the topic includes, *(i)* exploring the tool performance impact on encrypted DoH and DoT protocols, *(ii)* increasing the selection of tools being compared, and *(iii)* adding diversity of vantage points (*e.g.*, in different countries) and network conditions (*e.g.*, mobile networks) of the measurements.

References

Affinito, A., Botta, A., and Ventre, G. (2022). Local and Public DNS Resolvers: Do You Trade Off Performance Against Security? In *IFIP Networking Conference (IFIP Networking 2022)*, pages 1–9, Catania, Italy.

Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. (2010). Comparing DNS Resolvers in the Wild. IMC '10, page 15–21, New York, NY, USA. Association for Computing Machinery.

Astrand, P. (2003). PEP 324 – subprocess - New Process Module. `https://peps.python.org/pep-0324/`.

Borgolte, K., Chattopadhyay, T., Feamster, N., Kshirsagar, M., Holland, J., Hounsel, A., and Schmitt, P. (2019). How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem. *SSRN Electronic Journal*.

Böttger, T., Cuadrado, F., Antichi, G., Fernandes, E. L. a., Tyson, G., Castro, I., and Uhlig, S. (2019). An Empirical Study of the Cost of DNS-over-HTTPS. In *Proceedings of the Internet Measurement Conference (IMC 2019)*, pages 15—21, Amsterdam, Netherlands.

Bozkurt, I., Aguirre, A., Chandrasekaran, B., Godfrey, P., Laughlin, G., Maggs, B., and Singla, A. (2017). Why Is the Internet so Slow?! pages 173–187.

Butkiewicz, M., Madhyastha, H. V., and Sekar, V. (2011). Understanding Website Complexity: Measurements, Metrics, and Implications. In *ACM Conference on Internet Measurement Conference (IMC 2011)*, page 313–328, Berlin, Germany.

Chhabra, R., Murley, P., Kumar, D., Bailey, M., and Wang, G. (2021). Measuring DNS-over-HTTPS Performance around the World. In *ACM Internet Measurement Conference (IMC 2021)*, page 351–365, Virtual Event.

Doan, T. V., Tsareva, I., and Bajpai, V. (2021). Measuring dns over tls from the edge: Adoption, reliability, and response times. In Hohlfeld, O., Lutu, A., and Levin, D., editors, *Passive and Active Measurement*, pages 192–209, Cham. Springer International Publishing.

Hounsel, A., Borgolte, K., Schmitt, P., Holland, J., and Feamster, N. (2020). Comparing the effects of dns, dot, and doh on web performance. In *Proceedings of The Web Conference 2020*, WWW '20, page 562–572, New York, NY, USA. Association for Computing Machinery.

Hounsel, A., Schmitt, P., Borgolte, K., and Feamster, N. (2021). Can Encrypted DNS Be Fast? In *Passive and Active Measurement*, pages 444–459, Cham. Springer International Publishing.

Kurose, J. F. and Ross, K. W. (2016). *Computer Networking: A Top-Down Approach*. Pearson, 7 edition.

Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., and Joosen, W. (2019). Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed System Security Symposium (NDSS 2019)*, San Diego, USA.

Liu, C. and Albitz, P. (2001). *DNS and BIND*. O'Reilly & Associates, 4 edition.

Mockapetris, P. and Dunlap, K. J. (1988). Development of the Domain Name System. In *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM 1988)*, page 123–133, Stanford, California, USA.

Python Software Foundation (2023). `time` — Time Access and Conversions. `https://docs.python.org/3/library/time.html`.

Sharma, R., Feamster, N., and Hounsel, A. (2022). Measuring the Availability and Response Times of Public Encrypted DNS Resolvers. arXiv 2208.04999, cs.CR.

All links visited on 25/09/2023