Trois très très simples résolveurs DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 février 2022. Dernière mise à jour le 24 février 2022

https://www.bortzmeyer.org/tiny-resolver.html

Inspiré par un excellent article de Julia Evans <https://jvns.ca/blog/2022/02/01/a-dns-resolver-in-80->, voici trois très simples (vraiment très simples !) résolveurs DNS <https://www.bortzmeyer.org/resolveur-dns.html> en Python, en Elixir et en Rust. Ne les utilisez pas en production, leur but est purement pédagogique.

Comme dans l'article originel <https://jvns.ca/blog/2022/02/01/a-dns-resolver-in-80-lines-of-go>, l'idée est de montrer, via un programme, le fonctionnement d'un résolveur DNS <https://www.bortzmeyer.org/resolveur-dns.html>, notamment la **délégation** d'un domaine vers un sous-domaine, l'un des principes fondamentaux du DNS. Voici par exemple le déroulé de la résolution du nom www.st-cyr.terre.defense.gouv.fr (choisi pour son nombre de composants) :

```
% ./tiny-resolver.py www.st-cyr.terre.defense.gouv.fr A
Querying www.st-cyr.terre.defense.gouv.fr./A at 2001:7fd::1, depth 0
Querying www.st-cyr.terre.defense.gouv.fr./A at 2001:678:c::1, depth 1
Querying www.st-cyr.terre.defense.gouv.fr./A at 84.96.147.1, depth 2
Value of www.st-cyr.terre.defense.gouv.fr./A is ('OK', <DNS IN A rdata: 77.129.200.34>)
```

Trois serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html> ont été interrogés successivement, un de la racine, un de l'Afnic et un de SFR.

Ces deux résolveurs ont beaucoup de limites : ils ne prennent que le premier serveur faisant autorité pour chaque délégation (un vrai résolveur essaie au hasard, puis mémorise le serveur le plus rapide à répondre), par exemple. Et ils sont peu robustes face aux imprévus. Le deuxième résolveur ne gère même pas les alias (mais celui de Julia Evans non plus).

Le premier résolveur est écrit en Python. Il est téléchargeable en (en ligne sur `https://www.bortzmeyer.org/files/tiny-resolver.py`). Il dépend de la bibliothèque dnspython <`https://www.dnspython.org/`>. Si vous ne connaissez pas Python, cette bibliothèque peut s'installer avec, au choix, le système de paquetage de votre système d'exploitation (`apt install python3-dnspython` sur Debian) ou bien avec l'outil pip (`pip3 install dnspython`).

Le deuxième résolveur est écrit en Elixir. Il est téléchargeable en (en ligne sur `https://www.bortzmeyer.org/files/tiny-resolver-elixir.tar`). Elixir ne permet pas facilement de produire un exécutable avec un seul fichier dès qu'on utilise des bibliothèques extérieures comme elixir-dns <`https://github.com/tungd/elixir-dns`>. La marche à suivre est donc (ne pas oublier d'installer les sources d'Erlang - `erlang-src` sur Debian - pour pouvoir compiler la bibliothèque DNS) :

```
tar xvf tiny-resolver-elixir.tar
cd tiny-resolver-elixir
mix deps.get  # Et accepter l'installation de Hex, la source des paquetages
mix run resolver.exs domain-name
```

Quant au troisième, écrit par Kim Minh Kaplan <`https://github.com/kmkaplan/tiny-resolver-rs`>, il est en Rust. Il est téléchargeable en (en ligne sur `https://www.bortzmeyer.org/files/tiny-resolver.rs`). Vous devez avoir un Rust très récent et ajouter ce manifeste décrivant les dépendances, dans `Cargo.toml` :

```
[package]
name = "tiny-resolver"
version = "0.1.0"
edition = "2021"

[dependencies]
trust-dns-client = "0.20.4"
```

Ensuite, un `cargo build`, et vous trouverez le programme exécutable en `./target/debug/tiny-resolver` :

```
% ./target/debug/tiny-resolver www.afnic.fr
dig -r @198.41.0.4 www.afnic.fr
  d.nic.fr. 172800 IN A 194.0.9.1
dig -r @194.0.9.1 www.afnic.fr
  ns3.nic.fr. 172800 IN A 192.134.0.49
dig -r @192.134.0.49 www.afnic.fr
  www.afnic.fr. 600 IN A 91.188.78.66
Result: 91.188.78.66
```

Bonne lecture de code source !