Pourquoi ne pas mélanger résolveur DNS et serveur DNS faisant autorité ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 novembre 2019

https://www.bortzmeyer.org/separer-resolveur-autorite.html

Je donne souvent le conseil de ne pas configurer un serveur DNS à la fois en résolveur <https://www.bortzmeyer.org/resolveur-dns.html> et en serveur faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html>. Mais pourquoi, au juste ?

D'abord, fixons la terminologie. Parler de « serveur DNS » tout court (ou, encore pire de « DNS » pour désigner un serveur, du genre « pour être sûr que la NSA ait mes données, j'utilise 8.8.8.8 comme DNS ») est générateur de confusion. En effet, les deux types de serveurs DNS, **résolveurs <https://www.bortzmeyer.org/resolveur-dns.html>** et **serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html>** sont très différents, posent des questions bien diverses, et il est rare qu'on ait besoin de parler des deux ensemble. Quand quelqu'un écrit ou dit « serveur DNS », il s'agit en général de l'un ou de l'autre type, et cela vaudrait donc la peine de le préciser. Les termes de **résolveur** et de **serveur faisant autorité** sont bien définis dans le RFC 9499 [1] mais il est sans doute utile de rappeler ces définitions et quelques explications ici :

— Un résolveur (en anglais "*resolver*") est un serveur DNS qui ne connait presque rien au démarrage, mais qui va interroger, pour le compte du client final, d'autres serveurs jusqu'à avoir une réponse qu'il transmettra au client final. Dans une configuration typique, le résolveur que vous utilisez est géré par votre FAI ou bien par le service informatique de l'organisation où vous travaillez. (Il existe aussi des résolveurs publics, comme ceux de Cloudflare ou de Quad9 <https://www.bortzmeyer.org/quad9.html>.)
— Un serveur faisant autorité (en anglais "*authoritative server*", parfois bêtement traduit par « serveur autoritaire ») connait (« fait autorité pour ») une ou plusieurs zones DNS et il sert l'information sur ces zones aux résolveurs qui l'interrogeront. Il est typiquement géré par un hébergeur DNS ou par le bureau d'enregistrement ou directement par le titulaire du nom de domaine.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc9499.txt

Par exemple, au moment de la rédaction de cet article, je suis dans un train, avec le Wi-Fi, et le résolveur annoncé est `10.26.0.4`, une machine gérée par la SNCF. Si je veux me connecter à `celsa.fr`, ma machine demandera au résolveur `10.26.0.4`, qui relaiera aux deux serveurs faisant autorité, `ns0.idianet.net` et `ns1.idianet.net` (Idianet étant l'hébergeur DNS choisi par le CELSA), puis me retransmettra leur réponse.

La plupart des logiciels permettant de faire un serveur DNS ne permettent de faire qu'un résolveur, ou bien qu'un serveur faisant autorité. Comme je l'ai dit, c'est logique puisque ce sont deux fonctions très différentes, avec des problèmes bien distincts :

— Le serveur faisant autorité est public (s'il est interne à une organisation, la question de la séparation entre résolveur et serveur faisant autorité est différente), donc il doit faire face à un Internet pas toujours sympathique,
— Le résolveur, lui, est privé (il est déconseillé d'avoir des résolveurs ouverts <`https://www.bortzmeyer.org/fermer-les-recursifs-ouverts.html`>, sauf si on est Google et qu'on sait ce qu'on fait, cf. RFC 5358),
— Le serveur faisant autorité a des temps de réponse connus, et une consommation de mémoire stable,
— Le résolveur, lui, ne sait pas à l'avance combien de temps prendront les requêtes, et il peut avoir à allouer des ressources variables,
— Et, bien sûr, le serveur faisant autorité connait exactement les données, alors que le résolveur les obtient de l'extérieur, ce qui diminue la confiance qu'on peut leur accorder.

Mais certains logiciels (comme BIND) permettent d'assurer les deux fonctions, et en même temps. Certains administrateurs système ont donc configuré une machine où le même démon est résolveur et serveur faisant autorité. C'est une mauvaise idée, et ces administrateurs système ont tort. Pourquoi ?

Le principal problème pratique est de débogage. En effet, si le même serveur DNS est à la fois résolveur et serveur faisant autorité, les réponses reçues par un client DNS peuvent être issues du résolveur ou directement des zones DNS gérées. Normalement, avec un logiciel bien fait, les données faisant autorité auront toujours priorité (c'est bien ce que veut dire « faire autorité ») et les données issues de la résolution ne les remplaceront jamais. C'est ainsi que se comporte BIND aujourd'hui, mais cela n'a pas toujours été le cas, et ce n'est toujours pas le cas pour d'autres logiciels. On voit ainsi parfois des données récupérées lors du processus de résolution (et donc moins fiables, surtout lors d'attaques type empoisonnement de cache) masquer les données faisant autorité.

La seconde raison pour laquelle le mélange est une mauvaise idée est plus abstraite : les deux fonctions, résolveur et autorité, sont très différentes conceptuellement, et beaucoup de problèmes pratiques avec le DNS viennent d'une ignorance des acteurs (pas seulement les utilisateurs, même les administrateurs systèmes et réseaux) de ces deux fonctions, de leur rôle respectif, et des problèmes pratiques auxquelles elles font face.

Enfin, pour les programmeurs, on notera que, si ces deux fonctions ont du code en commun (encodage et décodage des requêtes et réponses, par exemple, et qui peut donc avoir intérêt à être dans une bibliothèque commune), l'essentiel du programme est différent. Ainsi, un serveur faisant autorité est sans état : il peut répondre aux questions immédiatement, il ne dépend de personne d'autre, et peut donc travailler de manière synchrone. Le résolveur, lui, est forcément avec état car il doit mémoriser les requêtes en cours de résolution, résolution dont la durée dépend des serveurs faisant autorité extérieurs. Si vous voulez essayer, notez qu'écrire un serveur faisant autorité est un projet relativement simple, alors qu'un résolveur est **beaucoup** plus compliqué (notamment du point de vue de la sécurité). Contrairement au serveur faisant autorité, il dépend fortement de serveurs extérieurs. (Comme une partie du code est quand même commune, une bonne architecture est celle de Knot : le serveur faisant autorité <`https://www.knot-dns.cz/`> et le résolveur <`https://www.knot-resolver.cz/`> partagent la même bibliothèque libknot pour certaines fonctions de base mais, autrement, ce sont deux logiciels très différents.)

Pour finir, voici un exemple de la différence entre les réponses fournies par un résolveur (`127.0.0.1`, ma machine locale, je ne suis plus dans le train), et celles fournies par un serveur faisant autorité pour la zone en question (`ns0.idianet.net`). D'abord, le résolveur :

```
% dig A celsa.fr

; <<>> DiG 9.11.5-P4-5.1-Debian <<>> A celsa.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53605
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;celsa.fr. IN A

;; ANSWER SECTION:
celsa.fr. 83049 IN A 195.154.244.151

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Nov 19 20:09:28 CET 2019
;; MSG SIZE  rcvd: 53
```

La réponse ne fait pas autorité (il n'y a pas de *"flag"* AA - *"Authoritative Answer"*), et le TTL a été décrémenté depuis sa valeur originale (l'information a été tirée de la mémoire du résolveur). Maintenant, la réponse d'un serveur faisant autorité :

```
% dig @ns0.idianet.net A celsa.fr

; <<>> DiG 9.11.5-P4-5.1-Debian <<>> @ns0.idianet.net A celsa.fr
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3519
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;celsa.fr. IN A

;; ANSWER SECTION:
celsa.fr. 86400 IN A 195.154.244.151

;; AUTHORITY SECTION:
celsa.fr. 86400 IN NS ns0.idianet.net.
celsa.fr. 86400 IN NS ns1.idianet.net.

;; ADDITIONAL SECTION:
ns0.idianet.net. 86400 IN A 195.154.201.10
ns1.idianet.net. 86400 IN A 195.154.244.138

;; Query time: 4 msec
;; SERVER: 195.154.201.10#53(195.154.201.10)
;; WHEN: Tue Nov 19 20:09:34 CET 2019
;; MSG SIZE  rcvd: 132
```

Cette fois, la réponse fait autorité ("*flag*" AA) et le TTL est la valeur originale (ici, une journée). On notera également le temps de réponse plus court du résolveur, puisqu'ici, l'information était déjà dans sa mémoire. Si le résolveur avait été « froid » (pas d'information en mémoire), le temps de réponse aurait été plus long, et le TTL aurait eu sa valeur originale (car l'information venait juste d'être récupérée).

En conclusion, il est recommandé de séparer les deux fonctions, de résolveur et de serveur faisant autorité. Même si ça semble marcher lors de la mise en service, mélanger les deux fonctions vous vaudra des maux de tête plus tard, lorsqu'un problème surviendra.