Le développement d'un serveur DNS, et les pièges de l'Internet (1/12)

Stéphane Bortzmeyer
stephane+cdl@bortzmeyer.org

Capitole du Libre, 19 novembre 2022

Le résumé

Le résumé

- Un logiciel libre,

Le résumé

- Un logiciel libre,
- Serveur DNS faisant autorité,

Le résumé

- Un logiciel libre,
- Serveur DNS faisant autorité,
- Orienté vers des réponses dynamiques,

Le résumé

- Un logiciel libre,
- Serveur DNS faisant autorité,
- Orienté vers des réponses dynamiques,
- Écrit en Elixir,

Le résumé

- Un logiciel libre,
- Serveur DNS faisant autorité,
- Orienté vers des réponses dynamiques,
- Écrit en Elixir,
- `https://framagit.org/bortzmeyer/drink`

Petit rappel sur Elixir

Petit rappel sur Elixir

- Langage fonctionnel,

Petit rappel sur Elixir

- Langage fonctionnel,
- Compilé en *bytecode* Erlang et exécuté par la VM Erlang,

Petit rappel sur Elixir

- Langage fonctionnel,
- Compilé en *bytecode* Erlang et exécuté par la VM Erlang,
- Parallélisme massif,

Petit rappel sur Elixir

- Langage fonctionnel,
- Compilé en *bytecode* Erlang et exécuté par la VM Erlang,
- Parallélisme massif,
- Accent mis sur la sécurité et la robustesse.

La motivation

La motivation

- Apprendre Elixir via un projet réel,

La motivation

- Apprendre Elixir via un projet réel,
- Trouver l'adresse IP du résolveur (les services qui le font disparaissent vite...)

La motivation

- Apprendre Elixir via un projet réel,
- Trouver l'adresse IP du résolveur (les services qui le font disparaissent vite...)
- Avoir une plate-forme d'expérimentation (hackathons IETF).

Services fournis

Services fournis

- Adresse IP du résolveur,

Services fournis

- Adresse IP du résolveur,
- ECS envoyé par le résolveur,

Services fournis

- Adresse IP du résolveur,
- ECS envoyé par le résolveur,
- Autres (AS et pays du résolveur).

Les principes de conception

Les principes de conception

- Un processus Erlang par requête,

Les principes de conception

- Un processus Erlang par requête,
- Jeux de tests détaillés, pour la robustesse.

Parallélisme

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,
- Chaque requête UDP crée un processus,

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,
- Chaque requête UDP crée un processus,
- Chaque connexion TCP aussi (le programme n'est pas bloqué par un client lent),

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,
- Chaque requête UDP crée un processus,
- Chaque connexion TCP aussi (le programme n'est pas bloqué par un client lent),
- Chaque requête TCP aussi,

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,
- Chaque requête UDP crée un processus,
- Chaque connexion TCP aussi (le programme n'est pas bloqué par un client lent),
- Chaque requête TCP aussi,
- *Pipelining* **et** réponses *out-of-order* ! (RFC 7766, sections 6.2.1.1 et 7)

Parallélisme

- Rappel : la fabrication des réponses peut prendre du temps,
- Chaque requête UDP crée un processus,
- Chaque connexion TCP aussi (le programme n'est pas bloqué par un client lent),
- Chaque requête TCP aussi,
- *Pipelining* **et** réponses *out-of-order* ! (RFC 7766, sections 6.2.1.1 et 7)
- Si ça plante, ça limite les dégâts.

Tests

Tests

- L'Internet est une jungle,

Tests

- L'Internet est une jungle,
- Quand on écoute sur le port 53, on reçoit plein de trucs bizarres,

Tests

- L'Internet est une jungle,
- Quand on écoute sur le port 53, on reçoit plein de trucs bizarres,
- Exemple : le nombre indiquant la taille d'une section peut mentir,

Tests

- L'Internet est une jungle,
- Quand on écoute sur le port 53, on reçoit plein de trucs bizarres,
- Exemple : le nombre indiquant la taille d'une section peut mentir,
- Plein d'autres exemples dans le RFC 9267.

Elixir aide

Elixir aide

- Pas de pointeurs,

Elixir aide

- Pas de pointeurs,
- Variables immuables,

Elixir aide

- Pas de pointeurs,
- Variables immuables,
- Les opérations sur les listes, les valeurs binaires, etc, sont gardées : lire au-delà de la valeur est sûr.

Tests réseau

Tests réseau

- Faits depuis un programme Python, pour varier,

Tests réseau

- Faits depuis un programme Python, pour varier,
- Interroge le serveur et vérifie les réponses,

Tests réseau

- Faits depuis un programme Python, pour varier,
- Interroge le serveur et vérifie les réponses,
- Toute la chaine est testée.

Choix des dépendances

Choix des dépendances

- Le problème douloureux du choix des dépendances,

Choix des dépendances

- Le problème douloureux du choix des dépendances,
- Qu'est-ce que je fais moi-même et qu'est-ce que je sous-traite ?

Choix des dépendances

- Le problème douloureux du choix des dépendances,
- Qu'est-ce que je fais moi-même et qu'est-ce que je sous-traite ?
- Évaluer si une bibliothèque est maintenue ou pas,

Choix des dépendances

- Le problème douloureux du choix des dépendances,
- Qu'est-ce que je fais moi-même et qu'est-ce que je sous-traite ?
- Évaluer si une bibliothèque est maintenue ou pas,
- Deux exemples : analyse DNS et réseau.

Performances

Performances

- `dnsperf -n 10000 -c 100 -s 127.0.0.1 -p 3553 -d data`

Performances

- `dnsperf -n 10000 -c 100 -s 127.0.0.1 -p 3553 -d data`
- Sur la même machine (12 cœurs à 1,5 GHz),

Performances

- `dnsperf -n 10000 -c 100 -s 127.0.0.1 -p 3553 -d data`
- Sur la même machine,
- NSD 4.6.0 : 152 000 r/s,

Performances

- `dnsperf -n 10000 -c 100 -s 127.0.0.1 -p 3553 -d data`
- Sur la même machine,
- NSD 4.6.0 : 152 000 r/s,
- Drink : 49 000 r/s (beaucoup moins si on active le *logging*),

Performances

- `dnsperf -n 10000 -c 100 -s 127.0.0.1 -p 3553 -d data`
- Sur la même machine,
- NSD 4.6.0 : 152 000 r/s,
- Drink : 49 000 r/s,
- Comparaison évidemment limitée (NSD ne sert que du contenu statique).