GRONG, un serveur de noms écrit en Go

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 février 2010. Dernière mise à jour le 15 février 2010

https://www.bortzmeyer.org/dnsserver-en-go.html

J'ai déjà parlé ici de mes débuts <https://www.bortzmeyer.org/go-langage.html> avec le langage de programmation Go. Ce langage, comme C ou C++ est adapté à des programmes système ou réseaux. Je viens donc de commencer un (très simple) serveur DNS en Go.

Il se nomme GRONG <http://github.com/bortzmeyer/grong> pour « *"Gross and ROugh Nameserver written in Go"* » (et n'a rien à voir avec la ville du même nom). Il est hébergé sur github <https://www.bortzmeyer.org/github.html>. Ne comptez pas utiliser GRONG en production à la place de BIND ou de nsd ! Il est plus lent mais, surtout, il n'a aucune protection contre les paquets DNS erronés. Si un méchant génère des paquets anormaux avec Scapy et les envoie à GRONG, celui-ci plante complètement.

GRONG a plutôt pour but d'explorer ce qu'on peut faire avec Go. J'ai particulièrement apprécié les goroutines <http://golang.org/doc/effective_go.html#goroutines> qui permettent d'exprimer très facilement le parallélisme, ce qui est utile pour un serveur réseau, qui doit pouvoir répondre à une requête alors que la précédente n'est pas terminée. Comme mon but était d'apprendre, j'ai tout refait moi-même et pas utilisé les paquetages DNS du module "net" <http://golang.org/pkg/net/>.

GRONG est uniquement un serveur DNS faisant autorité, il ne gère pas la récursion. Il est séparé en deux parties, un frontal qui analyse les requêtes entrantes et génère des requêtes sortantes correctes. Et un dorsal qui produit la réponse à partir de la requête. Plusieurs dorsaux différents sont possibles comme un serveur AS112 ou bien un serveur qui renvoie au client l'adresse IP du résolveur <https://www.bortzmeyer.org/trouver-adresse-ns.html>.

Aujourd'hui, les performances sont convenables mais on ne bat pas BIND, encore moins nsd. Par exemple, mesurées avec queryperf <https://www.bortzmeyer.org/performances-serveur-dns.html>, en local, sur une machine de bureau ordinaire :

```
% queryperf  -s ::1 -p 8053 -d 10.in-addr.arpa.queryperf

DNS Query Performance Testing Tool
Version: $Id: queryperf.c,v 1.12 2007/09/05 07:36:04 marka Exp $

[Status] Processing input data
[Status] Sending queries (beginning with ::1)
[Status] Testing complete

Statistics:

  Parse input file:     once
  Ended due to:         reaching end of file

  Queries sent:         100000 queries
  Queries completed:    100000 queries
  Queries lost:         0 queries
  Queries delayed(?):   0 queries

  RTT max:              0.129288 sec
  RTT min:              0.000873 sec
  RTT average:          0.019942 sec
  RTT std deviation:    0.013558 sec
  RTT out of range:     0 queries

  Percentage completed: 100.00%
  Percentage lost:        0.00%

  Started at:           Mon Feb 15 21:53:29 2010
  Finished at:          Mon Feb 15 21:55:09 2010
  Ran for:              99.875524 seconds

  Queries per second:   1001.246311 qps
```

Mais c'est à améliorer.