

# Analyser graphiquement l'affinité d'un client DNS pour un serveur

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 mai 2010

<https://www.bortzmeyer.org/affinity.html>

---

À la réunion OARC <<https://www.dns-oarc.net/>> de Prague le premier mai, la vidéo spectaculaire (sans laquelle les réunions sont tristes) était due à Duane Wessels (Verisign) qui a cherché à représenter graphiquement l'**affinité** des clients DNS pour les serveurs de la racine. L'idée est de mettre les serveurs dans un espace 3D. Le client qui envoie des requêtes à tous les serveurs équitablement est placé au centre de l'espace. Plus il a d'affinité pour un serveur (plus il lui envoie de requêtes), plus il se déplace en direction de ce serveur.

Un résolveur DNS normal (par exemple BIND), choisit le serveur le plus rapide de la liste disponible. Ainsi, la racine ayant treize serveurs, BIND les interroge tous, puis se souvient du plus rapide, et lui reste fidèle. Par contre, lorsqu'il faut choisir entre les différentes instances d'un nuage "*anycast*", le résolveur ne voit pas les différentes instances et c'est BGP qui fait tout.

La présentation ne consistait pas seulement en images fixes, l'animation permet de voir le déplacement des clients, par exemple lorsqu'une instance d'un nuage "*anycast*" stoppe, les clients se déplacent vers les autres (en moins de deux minutes, merci, BGP).

Le code et les données sont disponibles publiquement, en <<http://www.verisignlabs.com/~wessels/Affinity/>>. Mais attention, ce n'est pas forcément facile à interpréter. Pour vous donner une idée, vous pouvez commencer par voir les transparents <<https://www.dns-oarc.net/files/workshop-201005/2010-05-02-OARC-CENTR.pdf>> puis par regarder deux films tournés pendant la présentation, en <<http://www.youtube.com/watch?v=pIIFmYvx8Dc>> et <<http://www.youtube.com/watch?v=Gsr0HYOLk2w>>. Ensuite, si vous êtes alléchés, vous pouvez vous lancer dans la compilation et l'exécution.

Le tout a été développé en OpenGL (et c'est très convaincant, comme exemple qui donne envie d'utiliser ce système) avec les traces DNS des requêtes aux serveurs racine. D'abord, pour compiler, en prenant l'exemple d'un système Debian, il vous faudra les bibliothèques OpenGL (*libglu1-mesa-dev*). Ensuite, vous éditez le *Makefile* (par défaut, les variables sont pour MacOS, j'ai donc dû les commenter suivant les commentaires) et vous tapez *make*.

Une fois que cela compile, la syntaxe générale pour l'utilisation est :

```
affinity geom.dat frame.dat < dnsquery.dat
```

`geom.dat` étant les coordonnées des serveurs DNS dans l'espace, et `frame.dat` le cadre sur lequel ils sont dessinés (par exemple une carte du monde). `dnsquery.dat` contient les requêtes (c'est un **gros** fichier, les serveurs racine en voient, du trafic).

Mais, en pratique, il vaut mieux suivre les suggestions de l'auteur et donner quelques valeurs différentes aux paramètres. La démonstration à l'OARC a été faite avec :

```
affinity -c 0.6 -d 1.5 -i 0.1 -r 10 ...
```

Voyons quelques exemples. D'abord, avec l'ensemble des serveurs racines. Les données sont dans le répertoire `data/Roots-20100323`. Prenons le fichier `roots-dodec.dat` comme cadre (comme il y a plus de huit serveurs racine, `roots-cube.dat` ne donne pas de bons résultats) et `20100324.125000.dat.bz2` comme données. Une fois le téléchargement fait, lançons le tout (on décompresse à la volée car les fichiers sont gros et on n'a pas forcément envie de les stocker) :

```
% bunzip2 -c 20100324.125000.dat.bz2 | \
  ./affinity -c 0.6 -d 1.5 -i 0.1 -r 10 -w roots-dodec.dat
```

On voit alors une jolie image. Presser 's' la met en route et un deuxième 's' lui donne sa vitesse de croisière. On peut alors faire tourner le dodécaèdre et admirer la fidélité des clients de `I.root-servers.net` (alors que `G.root-servers.net` n'attire pas grand'monde).

Essayons ensuite avec les instances d'un nuage "anycast". Les données sont en `/data/A-20100209`. Prenons le fichier de placement des serveurs `a-geo.dat` et le fichier de données `20100210.180000.dat.bz2`.

```
bunzip2 -c 20100210.180000.dat.bz2 | \
  ./affinity -c 0.6 -d 1.5 -i 0.1 -r 10 -w a-geo.dat world_borders.dat
```

affichera alors l'affinité des clients des instances de `A.root-servers.net`. À mon avis, c'est en se plaçant au dessus du Pôle Nord qu'on a la meilleure vue. On voit par exemple les clients du nœud de Francfort qui sont tentés de temps à temps par celui de New York. Après, il faut passer du temps et explorer.

`A.root-servers.net` n'a pas beaucoup d'instances. On peut essayer avec `J.root-servers.net`, dans le répertoire `/data/J-20100323` mais il a plutôt le problème inverse et l'animation est donc assez confuse.

En gros, la conclusion de ces images est que les clients DNS sont très fidèles à une instance particulière d'un nuage anycast (BGP est finalement plutôt stable et, dans ce cas, le client ne contrôle pas quelle instance il utilise) alors que les clients sont plus volages en ce qui concerne le serveur racine utilisé (attention, la visualisation a des pièges, le centre étant plus petit, il donne l'impression d'une forte densité et que les clients y sont plus nombreux, sans doute faudrait-il une échelle logarithmique pour compenser cet effet).

Merci à Marco Davids pour l'aide.

Sur certains systèmes, il faut apparemment appliquer le "patch" suivant :

```
266,267c266,267
<     gluPerspective(50.0, (float) width / height, 1, 1024);
<     gluLookAt(0, 0, 4, 0, 0, 0, 0.0, 1.0, 0.0);
---
>     //gluPerspective(50.0, (float) width / height, 1, 1024);
>     //gluLookAt(0, 0, 4, 0, 0, 0, 0.0, 1.0, 0.0);
```