

# Quel est le plus rapide serveur DNS d'un groupe ?

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 février 2008

<https://www.bortzmeyer.org/le-plus-rapide-dns.html>

---

Mesurer les performances d'un service informatique est un art difficile mais néanmoins indispensable à l'administrateur système. Parmi ces services, le DNS présente la particularité que plusieurs serveurs répondent pour une zone donnée. De même, une machine connectée à l'Internet a souvent le choix entre plusieurs serveurs comme résolveurs. Dans un groupe, quel est le plus rapide de ces serveurs ? Un petit programme permet de répondre à cette question.

Supposons que je m'interroge sur le temps de réponse d'un service dont le nom figure dans la zone `example.com`. Le temps de réponse du DNS doit être pris en compte. Ou bien je me demande si je dois utiliser les résolveurs fournis par mon FAI, ou bien un BIND que j'ai installé sur ma machine, ou bien encore un service tiers comme OpenDNS, qui résoud gratuitement des noms de domaine en échange d'une manipulation des réponses pour insérer de la publicité <<https://www.bortzmeyer.org/opendns-non-merci.html>>. Quel résolveur est le plus rapide ?

On peut mesurer le temps de réponse d'un serveur avec `echoping` <<http://echoping.sourceforge.net/>> :

```
% echoping -n 3 -m dns c.nic.fr -t SOA fr
Elapsed time: 0.118128 seconds
Elapsed time: 0.074891 seconds
Elapsed time: 0.047331 seconds
---
Minimum time: 0.047331 seconds (5409 bytes per sec.)
Maximum time: 0.118128 seconds (2167 bytes per sec.)
Average time: 0.080116 seconds (3195 bytes per sec.)
Standard deviation: 0.029137
Median time: 0.074891 seconds (3418 bytes per sec.)
```

Ou bien on peut utiliser `dig` qui affiche, tout à la fin, le temps écoulé :

```
% dig @c.nic.fr SOA fr.
...
;; Query time: 55 msec
```

Mais pour comparer N serveurs, c'est un peu pénible. D'autant plus que dig, contrairement à echo-ping, ne permet pas de répéter un test, ce qui est indispensable lorsqu'on fait des mesures sur Internet, où les temps de réponse varient souvent considérablement.

Joe Abley a donc développé pour cela `qtest`, le programme décrit plus bas, programme que j'ai un peu amélioré. Ce programme prend comme argument une requête DNS par exemple `MX elysee.fr.` ou bien `ANY www.example.com.` et une liste de serveurs à tester. Il affiche les N serveurs les plus rapides (N vaut 1 par défaut mais on peut le changer avec l'option `-n`) ainsi que leur temps de réponse en milli-secondes. Par exemple :

```
% qtest "SOA fr." a.nic.fr c.nic.fr e.nic.fr
46 c.nic.fr/192.134.0.129
```

Ce test permettra de comparer les temps de réponse de ces trois serveurs de l'AFNIC (bien sûr, le résultat dépendra de l'endroit où on se trouve, d'autant plus que `e.nic.fr` est à la Réunion). Pour tester pour une zone donnée (ici, la racine), on peut utiliser la possibilité qu'à le shell d'exécuter une commande et d'utiliser son résultat comme variable, ici avec la syntaxe `$(commande arguments...)` :

```
% qtest "SOA ." $(dig +short NS .)
3 F.ROOT-SERVERS.NET./2001:500:2f::f
```

`f.root-servers.net` a répondu le premier, en trois milli-secondes. Pour voir davantage de serveurs :

```
% qtest -n 5 "SOA ." $(dig +short NS .)
3 F.ROOT-SERVERS.NET./2001:500:2f::f
3 M.ROOT-SERVERS.NET./2001:dc3::35
6 F.ROOT-SERVERS.NET./192.5.5.241
21 I.ROOT-SERVERS.NET./192.36.148.17
21 K.ROOT-SERVERS.NET./193.0.14.129
```

`qtest` est écrit en shell. Après analyse de ses arguments avec `getopt`, il boucle sur les serveurs de noms passés en argument. Pour chacun d'eux, il cherche l'ensemble des adresses IPv4 et IPv6 (les temps de réponse peuvent être très différents entre les deux protocoles), stocke chaque couple `nom de machine, adresse IP` dans une chaîne de caractères, la barre oblique servant de séparateur (le shell Unix n'a pas de structures de données perfectionnées). Pour chaque couple, `qtest` fait trois essais avec `dig`. Les calculs statistiques (moyenne, tri, hélas pas de médiane <<https://www.bortzmeyer.org/médiane-et-moyenne.html>>) sont faits avec `awk`.

Ici, on va comparer un résolveur local, ceux fournis par le FAI, et ceux de deux services commerciaux, OpenDNS et "*DNS Advantage*" de Neustar. Cette fois, on fait appel à des serveurs récursifs, ayant un cache des réponses précédemment obtenues, et le choix de la question est donc important. Si la réponse est dans le cache (ce qui est typiquement le cas si on demande l'adresse de `www.google.com`), tout ira bien plus vite! Essayons d'abord avec un nom qui est probablement dans le cache. `172.19.1.1` est un résolveur local, ceux commençant par `62.4` sont les résolveurs de mon FAI, ceux commençant par `208.67.222` sont à OpenDNS et ceux commençant par `156.154` sont ceux de "*DNS advantage*". `qtest` affiche toujours l'adresse IP après la barre oblique même lorsque, comme ici, c'est inutile puisque l'identité du serveur était déjà une adresse IP :

```
% qtest -n3 "A a.gtld-servers.net" 172.19.1.1 62.4.16.70 62.4.17.69 208.67.222.222 208.67.220.220 156.154.70.1 1
3 172.19.1.1/172.19.1.1
49 62.4.17.69/62.4.17.69
61 208.67.222.222/208.67.222.222
```

On voit que les résolveurs les plus proches sont les plus rapides (un test avec un nom inexistant, donc pas dans le cache, donne le même résultat) contrairement aux prétentions des vendeurs d'OpenDNS ou de Neustar. Il ne faut jamais se fier à la publicité. Le test a été fait depuis une machine en Europe, où OpenDNS et "*DNS Advantage*" ont peu de serveurs. Est-ce mieux aux États-Unis? Non (ici, les serveurs du fournisseur d'hébergement sont ceux commençant par 208.75 et 208.78 :

```
0 127.0.0.1/127.0.0.1
0 208.78.97.155/208.78.97.155
1 208.75.87.250/208.75.87.250
23 156.154.70.1/156.154.70.1
25 156.154.71.1/156.154.71.1
29 208.67.220.220/208.67.220.220
29 208.67.222.222/208.67.222.222
```

Bref, si on veut des temps de réponse rapides, rien ne vaut les serveurs de son FAI.

Vous pouvez récupérer le programme qtest (en ligne sur <https://www.bortzmeyer.org/files/qtest.sh>). Plus riche, plus graphique et plus complexe, il y a l'excellent Namebench <<http://code.google.com/p/namebench/>>.