

Mesurer les performances de serveurs DNS

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 Décembre 2007. Dernière mise à jour le 17
Décembre 2007

<http://www.bortzmeyer.org/performances-serveur-dns.html>

Avant de choisir logiciel ou matériel, si on est soucieux des performances, il faut **mesurer**. Comment mesurer les performances de serveurs de noms ?

Il existe plusieurs logiciels libres pour un serveur de noms faisant autorité (BIND, NSD, PowerDNS, etc) et chacun permet plusieurs configurations, par exemple BIND et PowerDNS peuvent utiliser une base de données MySQL ou PostgreSQL pour stocker les enregistrements DNS qu'ils vont servir. Quelle configuration est la plus rapide ? Et de combien ? Il est rare que les gens qui affirment très fort « X est plus rapide que Y » aient fait des mesures. Ne faisons pas comme eux, mesurons (<http://www.bortzmeyer.org/mesurer-temps-execution.html>). Nous utiliserons le client DNS **queryperf**, un programme de mesure de performance, qui est distribué avec BIND (dans le répertoire contrib/).

Sur une Debian, queryperf peut s'installer ainsi :

```
% apt-get source bind9
% cd bind9-9.4.2/contrib/queryperf
% ./configure
% make
# Pas de cible install dans le Makefile
% sudo cp queryperf /usr/local/sbin
```

Nous pouvons maintenant le lancer pour voir les options (je n'ai pas tout montré) :

```
Usage: queryperf [-d datafile] [-s server_addr] [-p port] [-q num_queries]
                [-b bufsize] [-t timeout] [-n] [-l limit] [-f family] [-l]
                [-i interval] [-r arraysize] [-u unit] [-H histfile]
                [-T qps] [-e] [-D] [-c] [-v] [-h]
-d specifies the input data file (default: stdin)
-s sets the server to query (default: 127.0.0.1)
-q specifies the maximum number of queries outstanding (default: 20)
-t specifies the timeout for query completion in seconds (default: 5)
...
```

L'option la plus importante est `-d`. `queryperf` prend les questions à interroger dans un fichier, qui a une syntaxe simple : une question par ligne, le nom de domaine et le type d'enregistrement DNS. Par exemple :

```
www.afnic.fr    A
postgresql.fr MX
```

On peut créer un tel fichier à la main mais ce n'est pas très pratique. Introduisons un programme, qui va être chargé de créer un fichier de questions « typiques ». (en ligne sur <http://www.bortzmeyer.org/files/gen-data-queryperf.py>) est un petit programme Python qui crée un fichier de questions. Ses options sont :

```
% python gen-data-queryperf.py -h
Usage: gen-data-queryperf.py [-n number] [-p percent-random] [-t TLD] [-m MAXSIZE] [-f zone-file]
```

L'option `-n` indique le nombre de questions produites. Essayons-le :

```
% python gen-data-queryperf.py -n 3
www.6fw84.org    A
www.wis4b6vfoi.org  A
www.t2cih3k.org  A
```

Avec un autre TLD :

```
% python gen-data-queryperf.py -n 3 -t example
www.0fjtvn3q0i.example  A
www.ngw7m3vq71.example  A
www.2ul.example        A
```

Par défaut, les noms de domaines sont générés aléatoirement. La plupart du temps, ils n'existent pas sur le serveur, qui répondra simplement `NXDOMAIN` (*"No Such Domain"*). Comme une telle réponse est plus rapide que d'extraire les données de la base (environ trois fois plus rapide avec `BIND`), cela peut donner une idée fautive des performances du serveur. Il vaut donc mieux générer des questions qui correspondent à la base du serveur. C'est à cela que sert l'option `-f` qui prend un fichier de zone en paramètre (`-p` sert à mettre quand même un peu de hasard dans les questions) :

```
% python gen-data-queryperf.py -t fx -n 1000000 -p 0.25 -f ~/tmp/fx.db > questions-fx
```

Nous disposons désormais d'un fichier de questions.

Lançons maintenant `queryperf` :

<http://www.bortzmeyer.org/performances-serveur-dns.html>

```
% queryperf -d questions-fx -s mon-serveur-a-tester
```

```
...
```

```
Statistics:
```

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

```
Queries sent:         1000000 queries
Queries completed:    1000000 queries
Queries lost:         0 queries
Queries delayed(?):   0 queries
```

```
RTT max:             0.177042 sec
RTT min:             0.000136 sec
RTT average:         0.001147 sec
RTT std deviation:   0.000691 sec
RTT out of range:    0 queries
```

```
Percentage completed: 100.00%
Percentage lost:      0.00%
```

```
Started at:          Fri Dec 14 18:17:17 2007
Finished at:         Fri Dec 14 18:18:15 2007
Ran for:             58.112542 seconds
```

```
Queries per second:  17207.989284 qps
```

On voit que cette machine (un quadri-Opteron avec 8 gigaoctets de mémoire, Linux 2.6.22 et BIND 9.4.2 sur un fichier de zone de 2,1 millions d'entrées - environ un million de sous-domaines) peut traiter 17 000 requêtes DNS par seconde, sans aucune perte. Le temps de réponse moyen est de 1,147 milliseconde mais avec un écart-type relativement important.

Sur la même machine, je change le logiciel pour NSD 2.3.7 et je compile ma zone (nsdcrebuild&&nsdcreload). queryperf, avec le même fichier de questions, me dit :

```
Statistics:
```

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

```
Queries sent:         1000000 queries
Queries completed:    1000000 queries
Queries lost:         0 queries
Queries delayed(?):   0 queries
```

```
RTT max:             0.001606 sec
RTT min:             0.000156 sec
RTT average:         0.000278 sec
RTT std deviation:   0.000045 sec
RTT out of range:    0 queries
```

```
Percentage completed: 100.00%
Percentage lost:      0.00%
```

```
Started at:          Fri Dec 14 18:32:36 2007
Finished at:         Fri Dec 14 18:32:52 2007
Ran for:             15.554970 seconds
```

```
Queries per second:  64288.134275 qps
```

D'où on peut raisonnablement conclure que NSD est nettement plus rapide que BIND (et plus régulier).

On pourrait mesurer bien d'autres choses :

<http://www.bortzmeyer.org/performances-serveur-dns.html>

-
- Comparer les systèmes d'exploitation et pas uniquement les serveurs de noms (certains systèmes sont très lents en UDP).
 - Comparer IPv4 et IPv6 (hypothèse non testée : IPv6 sera plus lent, car l'implémentation est moins optimisée). queryperf marche en IPv6.
 - Tester l'effet de certaines options DNS comme EDNS0 (décrite dans le RFC 2671¹), option `-e` ou DNSSEC, option `-D`.

Bien sûr, comme avec tous les "*benchmarks*", il y a de nombreuses limites à ces mesures. Les mesures de performances génèrent souvent de longues discussions car on peut toujours changer encore un paramètre. Voici une liste non-exhaustive d'améliorations possibles :

- Utiliser plusieurs clients queryperf simultanés, et pas forcément sur le réseau local. Cela serait plus réaliste.
- Tester avec d'autres programmes de mesure de performance. echoping (<http://echoping.sourceforge.net/>), depuis sa version 6, permet de mesurer les serveurs DNS (mais sans pouvoir les bombarder comme queryperf, echoping mesure plutôt des requêtes isolées). Nominum **semble** avoir une version améliorée de queryperf (cf. http://www.nominum.com/services/measurement_tools.php).
- Utiliser un fichier de **vraies** questions et pas de questions générées comme ici. En effet, rien ne garantit que le temps de réponse sera le même pour tous les noms. Il serait donc peut-être préférable, si on est déjà en production, de récupérer les questions effectivement posées (avec un outil comme dnscap (<http://public.oarci.net/tools/dnscap/>) pour capturer les requêtes et un petit script - que je n'ai pas encore écrit - ensuite pour les reformater pour queryperf).

Cet article concerne la mesure de la performance d'un logiciel serveur. Si on s'intéresse à une comparaison de plusieurs serveurs déjà installés, on peut regarder le programme qtest (<http://www.bortzmeyer.org/le-plus-rapide-dns.html>).

¹Pour voir le RFC de numéro NNN, <http://www.ietf.org/rfc/rfcNNN.txt>, par exemple <http://www.ietf.org/rfc/rfc2671.txt>