

Contents

1	Commandes de création de l'environnement de test	1
1.1	Dans l'environnement dns-g5k (a.k.a l'environnement du papier):	1
1.2	Sur la frontale de grid5000, avec une réservation.	1
1.3	Sur la machine elle même	2
1.3.1	Enregistrement de la configuration actuelle :	2
1.3.2	Installation de dnsmdist	2
1.3.3	Paramétrage de différentes limites du noyau	6
2	soft nofile 4000000	9
3	hard nofile 4000000	9
3.0.1	Installation de linux-perf	9
3.0.2	Installation des outils nécessaires à la prise de mesures	9

1 Commandes de création de l'environnement de test

1.1 Dans l'environnement dns-g5k (a.k.a l'environnement du papier):

On execute `dnsmdist --version` pour avoir le plus d'informations sur les options de build de dnsmdist.

```
dnsmdist 1.7.3 (Lua 5.1.4 [LuaJIT 2.1.0-beta3])
Enabled features: cdb dns-over-tls(gnutls openssl) dns-over-https(DOH) dnscrypt
ebpf fstrm ipcipher libsodium lmbdb outgoing-dns-over-https(nghttp2) protobuf re2
recvmmsg/sendmmsg snmp systemd
```

Toutes ces commandes ont été testées le jeudi 7 décembre 2023, à partir de 14h51.

1.2 Sur la frontale de grid5000, avec une réservation.

```
- ~kadeploy3 debiantesting-min~
```

1.3 Sur la machine elle même

1.3.1 Enregistrement de la configuration actuelle :

Dans un premier temps, on fait la liste des paquets installés sur l'image.

```
mkdir 'pkg-lists'
```

```
apt list --installed > 'pkg-lists/base-image-pkgs'
```

Pour être sûr d'être à jour :

```
apt update && apt dist-upgrade
```

```
apt list --installed > 'pkg-lists/base-image-pkgs-update'
```

On fait une première sauvegarde de l'environnement avec :

```
tgz-g5k -m $(head -n 1 "$OAR_NODEFILE") -f 'dnsdist-debian-testing.tar.zst'
```

On re-déploie afin de redémarrer :

```
kadeploy3 -a 'dnsdist-debian-testing.yaml'
```

On récupère la version du noyau (avec `uname -r` : 6.5.0-5-amd64).

1.3.2 Installation de dnsdist

Installation de `wget` :

```
apt install wget
```

```
apt list --installed > 'pkg-lists/pkg-lists/pkgs-after-wget-install'
```

Installation de `bzip2` :

```
apt install bzip2
```

```
apt list --installed > 'pkg-lists/pkgs-after-bzip2-install'
```

Installation de `build-essential` :

```
apt install build-essential
```

```
apt list --installed > 'pkg-lists/pkgs-after-build-essential-install'
```

Installation de `git` :

```
apt install git
apt list --installed > 'pkg-lists/pkgs-after-git-install'
```

Installation de man-db :

```
apt install man-db
apt list --installed > 'pkg-lists/pkgs-after-man-db-install'
```

On refait une sauvegarde de l'environnement :

```
tgz-g5k -m $(head -n 1 "$OAR_NODEFILE") -f 'dnsdist-debian-testing.tar.zst'
```

Installation de la version 1.8.2-3 de dnsdist des dépôts de debian (pour la version avec h2o pour http) :

```
apt install dnsdist
apt list --installed > 'pkg-lists/pkgs-after-dnsdist-install'
```

A noter qu'il n'y a pas de dépendances sur les paquets debian pour la lib h2o car elle est vendored pour le build debian (e.g., incluse dans les sources).

Sortie de dnsdist --version :

```
dnsdist 1.8.2 (Lua 5.1.4 [LuaJIT 2.1.0-beta3])
Enabled features:
cdb dns-over-tls(openssl) dns-over-https(DOH) dnscrypt ebpf fstrm ipcipher
libedit libsodium lmbd outgoing-dns-over-https(nghttp2) protobuf re2
recvmmsg/sendmmsg snmp systemd
```

1. Récupération des sources de dnsdist :

```
mkdir sources
cd sources
wget 'https://downloads.powerdns.com/releases/dnsdist-1.9.0-alpha3.tar.bz2'
tar -xvf dnsdist-1.9.0-alpha3.tar.bz2
mv dnsdist-1.9.0-alpha3 dnsdist-1.9.0-alpha3
cd dnsdist-1.9.0-alpha3
```

2. Installation des dépendances :

Installation de libsodium-dev:

```
apt install libsodium-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libsodium-dev-install"
```

Installation de pkg-config pour que dnsdist reconnaisse que j'ai installé libsodium-dev :

```
apt install pkg-config
apt list --installed > "$HOME/pkg-lists/pkgs-after-pkg-config-install"
```

Installation de libboost-all-dev:

```
apt install libboost-all-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libboost-all-dev-install"
```

Installation de libre2-dev:

```
apt install libre2-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libre2-dev-install"
```

Installation de libsnmp-dev:

```
apt install libsnmp-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libsnmp-dev-install"
```

Installation de libsystemd-dev:

```
apt install libsystemd-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libsystemd-dev-install"
```

Installation de liblua5.1-dev:

```
apt install liblua5.1-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-liblua5.1-dev-install"
```

Installation de liblua5.1-dev:

```
apt install libnghttp2-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libnghttp2-dev-install"
```

Installation de libcdb-dev:

```
apt install libcdb-dev
apt list --installed > "$HOME/pkg-lists/pkgs-after-libcdb-dev-install"
```

Avant d'installer la libh2o manuellement, fait une sauvegarde de l'environnement tel qu'il est actuellement:

```
tgz-g5k -m $(head -n 1 "$OAR_NODEFILE") -f 'dnsdist-debian-testing.tar.zst'  
cp dnsdist-debian-testing.tar.zst dnsdist-debian-testing-before-h2o.tar.zst
```

Installation de libh2o-evloop-dev:

```
apt install libh2o-evloop-dev  
apt list --installed > "$HOME/pkg-lists/pkgs-after-libh2o-evloop-dev-install"
```

Installation de libedit-dev:

```
apt install libedit-dev  
apt list --installed > "$HOME/pkg-lists/pkgs-after-libedit-dev-install"
```

Installation de liblmbd-dev:

```
apt install liblmbd-dev  
apt list --installed > "$HOME/pkg-lists/pkgs-after-liblmbd-dev-install"
```

Installation de libfstrm-dev:

```
apt install libfstrm-dev  
apt list --installed > "$HOME/pkg-lists/pkgs-after-libfstrm-dev-install"
```

```
./configure --with-cdb=yes --enable-dns-over-tls=yes \  
            --enable-dns-over-https=yes --enable-dnscrypt=yes \  
            --with-ebpf=yes --enable-ipcipher=yes \  
            --with-libsodium=yes \  
            --with-lmdb=yes --with-re2=yes --with-net-snmp=yes \  
            --enable-systemd=yes --with-nghttp2=yes --with-h2o=yes
```

On a enlevé des arguments de la commande `./configure` présentée ici le flag `--with-libedit=yes`, car l'activer desactive aparemment `libedit` (configure: libedit: no dans la sortie de la commande).

Après avoir compilé (`make -j 18`), on execute la commande `dnsdist --version` :

```
dnsmdist 1.9.0-alpha3 (Lua 5.1.4 [LuaJIT 2.1.0-beta3])
```

Enabled features:

```
cdb dns-over-tls(openssl) dns-over-https(h2o nhttp2) dnscrypt ebpf fstrm  
ipcipher libedit libsodium lmbd protobuf re2 recvmmsg/sendmmsg snmp systemd
```

On note l'absence de `outgoing-dns-over-https(nhttp2)` qui était présent dans la sortie de `dnsmdist --version` pour les versions 1.7.3 et 1.8.2 (la 1.8.2 étant issue des dépôts de debian et la 1.7.3 des dépôts apt de dnsmdist).

On explique cette absence par le fait que, dans les $\geq 1.9.0$, le support de HTTP est harmonisé dans la librairie `nhttp2` : avant cette version, le doh entrant était géré par la lib `h2o`, et le doh sortant par `nhttp2`. Dans les versions supérieures à la 1.9, tout ce qui concerne doh est géré par `nhttp2`. Il est cependant encore possible d'utiliser `h2o` pour le doh entrant en passant `--with-h2o=yes`, et en passant `library="h2o"` à la directive `addDoHLocal()` du fichier de configuration.

On note que `libedit` et `dnscrypt` sont présentes dans la version de debian, ainsi que dans celle compilée par nos soins.

On fait `make install` pour installer dnsmdist dans `/usr/local/bin/`. On a désormais deux versions de dnsmdist installées : une dans `/usr/bin/`, et une dans `/usr/local/bin/` (du fait des règles de précédence de `$PATH`, c'est la version dans `/usr/local/bin/` qui est exécutée par défaut lorsqu'un chemin absolu n'est pas précisé.

Vu qu'on a les deux versions correctement installées, on sauvegarde l'image :

```
tgz-g5k -m $(head -n 1 "$OAR_NODEFILE") -f 'dnsmdist-debian-testing.tar.zst'
```

1.3.3 Paramétrage de différentes limites du noyau

1. `/etc/sysctl.conf` (symlinké depuis `/etc/sysctl.d/99-sysctl.conf`)

```
# Custom defaults for high-performance networking
```

```
# Ports that can be automatically assigned by the kernel  
net.ipv4.ip_local_port_range = 2000 65535
```

```
# global maximum number of opened files
```

```

fs.file-max = 9223372036854775807

# default size (in bytes) of the socket receive and send buffers
net.core.rmem_default=2097152
net.core.wmem_default=2097152

# Maximum size (in bytes) of the socket send an receive buffers
# which may be set by using the SO_RCVBUF socket option:
# (2 megs)
net.core.rmem_max=104857600
net.core.wmem_max=104857600

# tcp_rmem - vector of 3 INTEGERS: min, default, max
# min: Minimal size of receive buffer used by TCP sockets. It is guaranteed to ea
# Default: 4K
# default: initial size of receive buffer used by TCP sockets. This value override
# max: maximal size of receive buffer allowed for automatically selected receiver
# Calling setsockopt() with SO_RCVBUF disables automatic tuning of that socket's r
net.ipv4.tcp_rmem = 1048576 2097152 6291456

# tcp_wmem - vector of 3 INTEGERS: min, default, max
# min: Amount of memory reserved for send buffers for TCP sockets. Each TCP socket
# Default: 4K
# default: initial size of send buffer used by TCP sockets. This value overrides r
# It is usually lower than net.core.wmem_default.
# Default: 16K
# max: Maximal amount of memory allowed for automatically tuned send buffers for T
# Calling setsockopt() with SO_SNDBUF disables automatic tuning of that socket's s
# net.ipv4.tcp_wmem = 1048576 2097152 6291456
# Not touched : we don't touch it yet.

# Minimal size of receive buffer used by UDP sockets in moderation.
# Each UDP socket is able to use the size for receiving data, even
# if total pages of UDP sockets exceed udp_mem pressure. The unit is
# byte.
# (100 megs)
net.ipv4.udp_rmem_min=26214400

# UDP does not have tx memory accounting, so the setting
# "net.ipv4.udp_wmem_min" does not have any effect

```

```

# We do not modify net.ipv4.udp_mem, as net.ipv4.udp_rmem_min
# overrides it

# somaxconn - tcp_syn_backlog

# According to https://www.kernel.org/doc/html/latest/networking/ip-sysctl.html :
# tcp_max_syn_backlog - INTEGER
# Maximal number of remembered connection requests (SYN_RECV), which have not
# received an acknowledgment from connecting client.
# This is a per-listener limit.
# The minimal value is 128 for low memory machines, and it will increase in
# proportion to the memory of machine.
# If server suffers from overload, try increasing this number.
# Remember to also check /proc/sys/net/core/somaxconn A SYN_RECV request
# socket consumes about 304 bytes of memory.

# somaxconn - INTEGER

# Limit of socket listen() backlog, known in userspace as SOMAXCONN. Defaults to 4
# (Was 128 before linux-5.4) See also tcp_max_syn_backlog for additional tuning fo
# sockets.

# Useful SO link :
# stackoverflow.com/questions/62641621/what-is-the-difference-between-tcp-max-syn-

# Basically the difference between somaxconn and tcp_max_syn_backlog is that somax
# is the maximum number of connexions that can be in the "accept queue" : that is
# maximum number of ESTABLISHED connection that have not been accepted. tcp_max_sy
# other hand, is the maximum number of connexions that are in the "SYN_RECV" state
# When a connection is in this state, this means that the kernel has received a "S
# with the corresponding "SYN+ACK" packet, but has not yet received the correspond
# client (e.g., all sockets are in this queue before going into the "accept queue"
# not-yet-established) connections will leave this queue).

net.core.somaxconn=8192
net.ipv4.tcp_max_syn_backlog=8192

# 'netdev_max_backlog' : maximum size of the receive queue.
# Received frames will be stored in this queue after taking them from the ring bu

```

```
net.core.netdev_max_backlog=65536

# Amount of ancillary data for each socket (in bytes)
net.core.optmem_max=65536

# tcp_tw_reuse - INTEGER
# Enable reuse of TIME-WAIT sockets for new connections when it is safe from prot
net.ipv4.tcp_tw_reuse=1
```

2. /etc/security/limits.conf

```
#+BEGINsrc
```

2 soft nofile 4000000

3 hard nofile 4000000

```
#+ENDsrc
```

3.0.1 Installation de linux-perf

Installation de linux-perf:

```
apt install linux-perf
apt list --installed > "$HOME/pkg-lists/pkgs-after-linux-perf-install"
```

3.0.2 Installation des outils nécessaires à la prise de mesures

python3 et procps sont déjà installés.

Pour la mesure de la fréquence, on utilise linux-cpupower .

Installation de linux-cpupower:

```
apt install linux-cpupower
apt list --installed > "$HOME/pkg-lists/pkgs-after-linux-cpupower-install"
```

Installation de psmisc:

```
apt install psmisc
apt list --installed > "$HOME/pkg-lists/pkgs-after-psmisc-install"
```

Installation de htop:

```
apt install htop  
apt list --installed > "$HOME/pkg-lists/pkgs-after-htop-install"
```

ON A RECOPIE LA VIEILLE VERSION DE SYSCTL.CONF DANS
L4IMAGE POUR TESTER