exa
networks

exa networks

Présentation d'ExaBGP

Auteur de ces diapositives

Expat vivant en Angleterre depuis 1999 (et même avant)

« Hello. » - « Are you french ? »
« Merci. » - « Votre français est très bon, monsieur ! »

Second FRnOG, ou devrais-je dire deuxième ?

Directeur et co-fondateur d'Exa Networks (FAI B2B et éducation)
Directeur à LINX (6 ans) et IXLeeds (depuis sa création en 2010)

Programmeur Python depuis le siècle dernier
Jitsuka, Aikidoka, …

Table des matières

1. Présentation « rapide » de BGP
2. Présentation du logiciel
3. Questions (Réponses optionnelles)
4. Profit ???
5. Présentation suivante

Proposition de CAPTCHA

BGP en un slide (et une image)

Echange d'information (OPEN)

- négociations de fonctionnalités
- Pas négocié, pas de chocolat

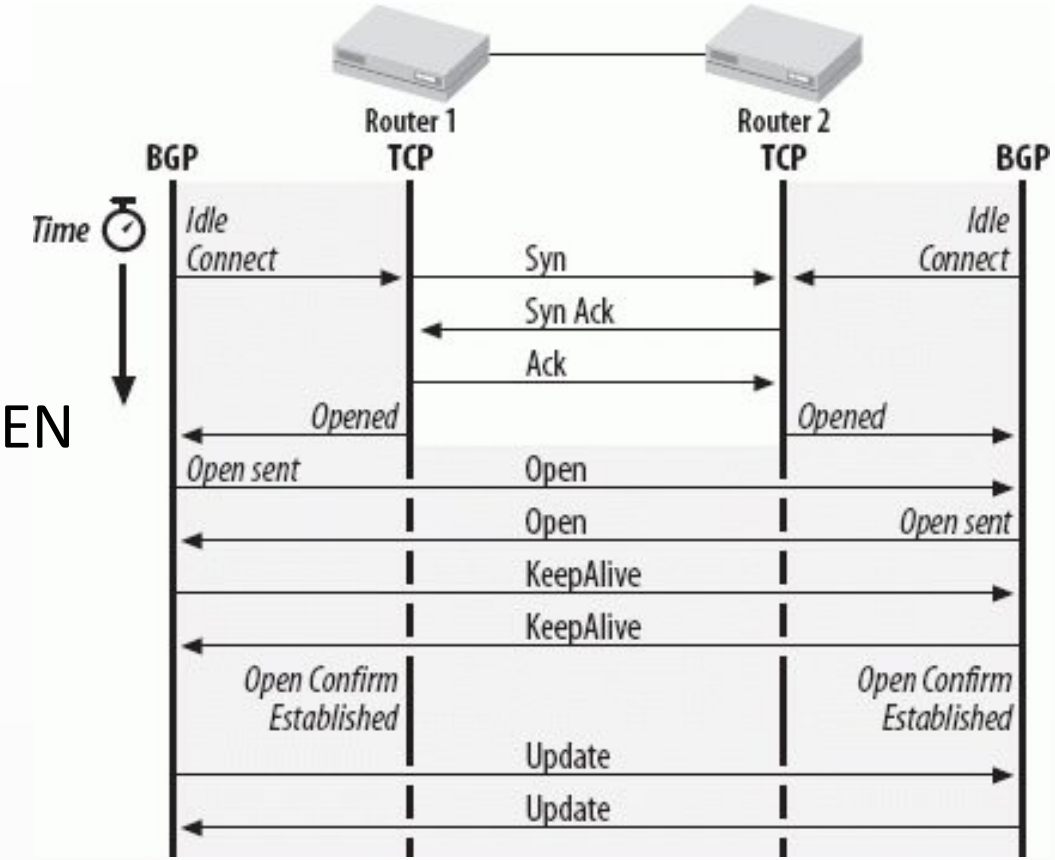
Echange d'un paquet de survie (KEEPALIVE)

- Intervalle d'échange issue des paquets OPEN

Echange de 'routes' (UPDATE)

- Caractéristiques (attributes –> extensible)
- 'Best path selection'

Un paquet d'erreur (NOTIFICATION)



ExaBGP, Pourquoi ?

« To scratch an itch »

- Première publication en Septembre 2009
- Serveur SVN, puis Google Code et enfin GitHub

```
# find lib -name "*py" | xargs cat | egrep -v "(^$|^\s*#.*)" | wc -l
First commit:      1,117
Master:           25,925   (commentaires 2461, lignes vides 5760)
```

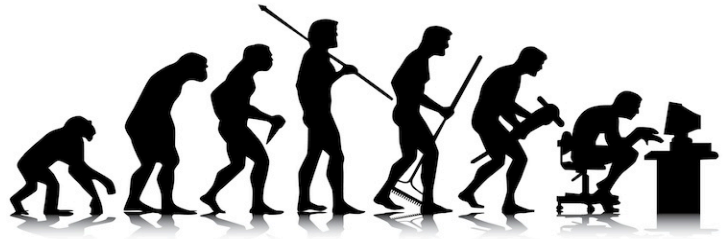
En python pour s'intégrer dans la « stack » d'Exa Networks

(Go n'existait pas, ceci dit Twitch et FranceIX non plus)

ExaBGP, Au fil du temps

Pas forcement exactement dans cet ordre

- Graceful Restart pour éviter « route flapping »
- Ajout du support « multiprotocol » (pour IPv6)
- Ajout de FlowSpec
- Ajout du contrôle par processus externe
- Décode les paquets reçus
- Acceptation des connexions entrantes
- … plus de 20 RFC (que je n'utilise pas) et quelques drafts



ExaBGP, l'Apache de BGP?

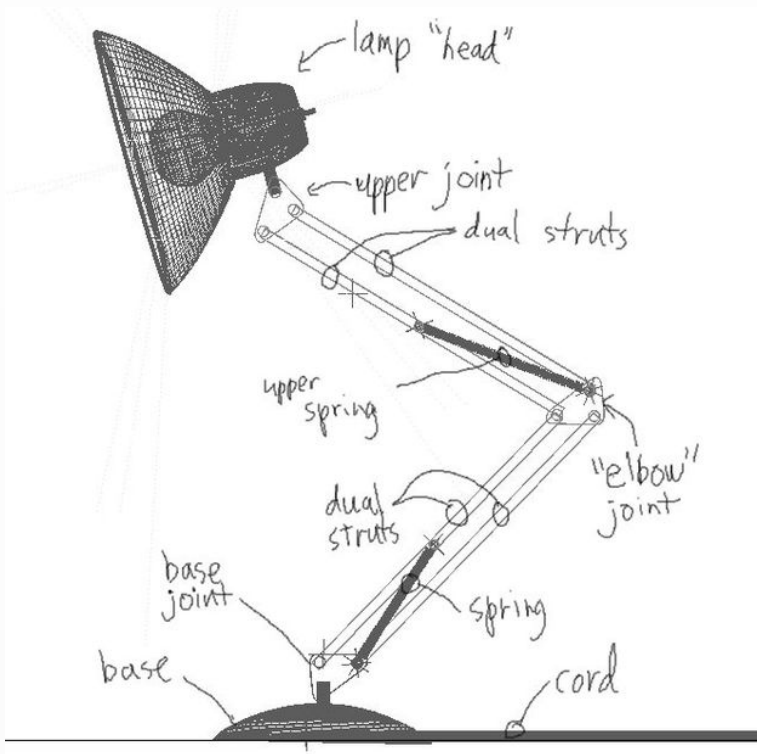
Serveur web – application – base de données

- Connu sous le petit nom de "LAMP stack"
- FreeBSD, NGINX, PostGreSQL, Python ..

HTTP est simple d'utilisation

```
newbie@home> cd ~/.ssh; python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 …
```

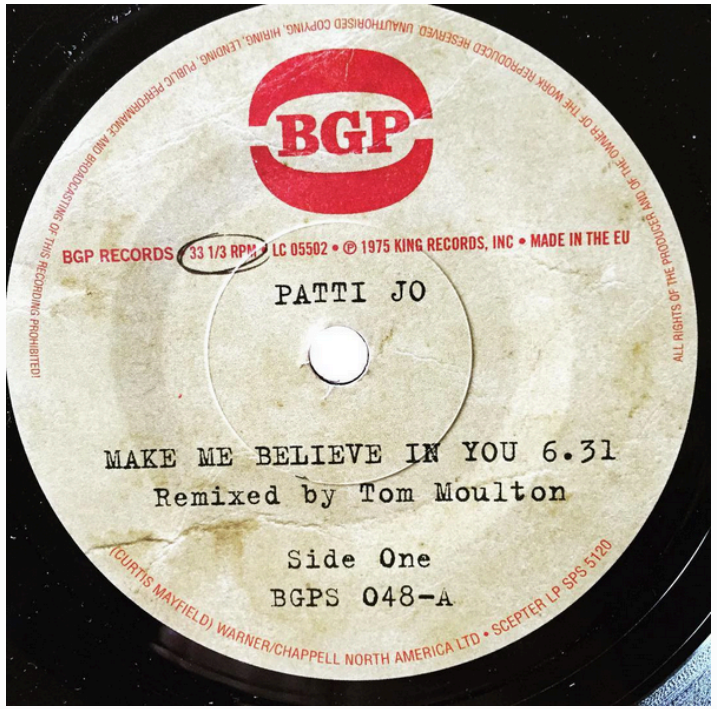
- BGP n'est pas plus compliqué que HTTP



ExaBGP, vue de haut ?

Un démon BGP

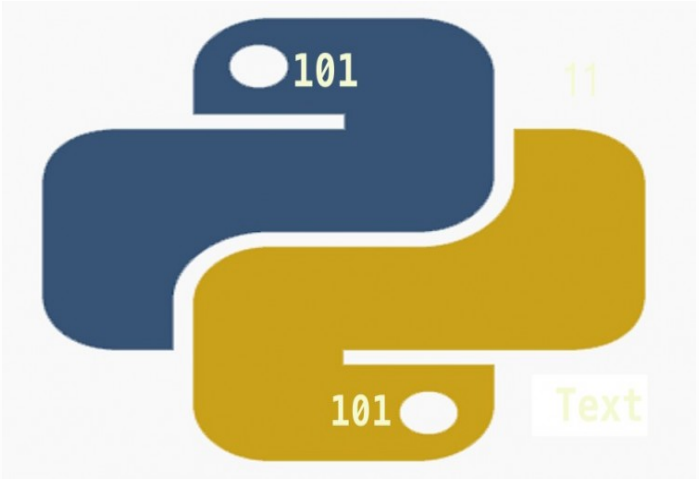
- Qui lit un fichier de configuration
- Qui lui indique où est un processus de contrôle
  - Exécuté par ExaBGP, redémarré en cas de faute
  - Un « CGI » qui ne s'arrête jamais
    - Communique avec ExaBGP
    - Annonce ou reçoit de routes
- Répond a plusieurs signaux (SIGUSR1, …)
- Maintient une "ADJ-RIB-OUT"
  - Pour re-envoyer les routes après deconnexion des peers (soumis à conditions)



ExaBGP, 0000 0101

exa networks

Installation




```
# pip install exabgp
# python -m exabgp healthcheck --help

# git clone https://github.com/Exa-Networks/exabgp.git
# cd exabgp
# git checkout 3.4
# export PATH=$PATH:`pwd`/exabgp/sbin

# exabgp --help
```

ExaBGP, command line

Options utiles en cas de problèmes

- -t vérification du fichier de configuration
- -1 une seule connexion TCP vers le peer
- -d active presque toutes les options de débogage
- -p en cas de crash, démarre le débogueur de python

ExaBGP reporte sur STDERR ce qu'il faut me fournir sur GitHub pour avoir de l'aide

SVP, SVP, SVP pour les reports de bugs sur GitHub délimitez le copier/coller avec ```

Si vous avec un compte Github, contactez-moi sur Gitter
https://gitter.im/Exa-Networks/exabgp

ExaBGP, Configuration

ExaBGP a un fichier de configuration (format .ini)

Principalement log, cache et option pour le démon

```
# ./sbin/exabgp --help  | grep -- "- exabgp." | wc –l
37

# env exabgp.log.all=1 ./sbin/exabgp --diff-ini
[exabgp.log]
all = true
```

-- diff-ini, --diff-env, --full-ini, --full-env

ExaBGP, SIGNAL

Les signaux suivants peuvent être utilisés:

- SIGLARM : redémarre ExaBGP
- SIGUSR1 : relit la configuration
- SIGUSR2 : relit la configuration et redémarre les programmes controlés
- SIGTERM : tue proprement ExaBGP
- SIGHUP : tue ExaBGP (integration avec “screen”)
  https://github.com/exa-networks/exabgp/issues/32

ExaBGP, moteur et performances

Boucle

jusqu'à une seconde (temps du reacteur)

« avance » chaque peer

regarde les IOS avec select, les traite une fois

ajoute les connexions entrantes

AsyncIO avec python 2.6/2.7, comme windows 3.1

python threads sont bloquantes (GIL)

2,000 peers par boucle/cœur (OpenBGPD ML)

> ./sbin/exabgp un.conf deux.conf trois.conf quatre.conf

ExaBGP démarre quatre ExaBGP, utilisant quatre cœurs

ExaBGP, fichier de configuration

```
neighbor 127.0.0.1 {
        description "a quagga test peer";
        router-id 10.0.0.2;
        local-address 127.0.0.1;
        local-as 65533;
        peer-as 65533;
        hold-time 180;

        static {
                route 10.0.1.0/24 {
                        next-hop 10.0.255.254;
                        community [ 0 ];
                        as-path [ 30740 30740 ];
                        local-preference 200;
                }
                route 10.0.2.0/24 next-hop 10.0.255.254 origin igp;
        }
}
```

Syntaxe très 'JunOS'

Absent de cet exemple

- familles
  - ipv6 unicast
  - ipv4 mpls-vpn
  - ipv6 flow
- capacités
  - route-refresh
  - graceful-restart
  - add-path send/receive

ExaBGP, décodeur BGP

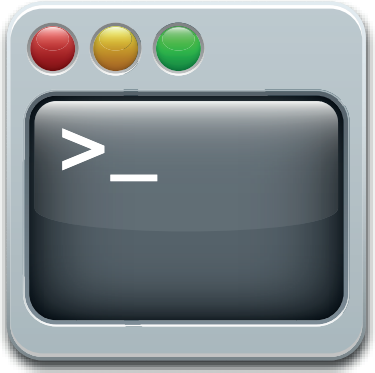
```
# ./sbin/exabgp \
        qa/conf/api-open.conf \
        --decode \
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF:003C:02:0000001C4001010040020040030465016501
800404000000C840050400000064000000002001010101  \
        2>&1 | egrep "(decoded|json)" | cut –b65-
decoded update 1 1.1.1.1/32 path-information 0.0.0.0 next-hop 101.1.101.1 origin igp med
200 local-preference 100
update json { "exabgp": "3.5.0", "time": 1460367619, "host" : "thomas.local", "pid" :
32511, "ppid" : 86579, "counter": 1, "type": "update", "neighbor": { "address": { "local":
"127.0.0.1", "peer": "127.0.0.1" }, "asn": { "local": "1", "peer": "1" }, "direction": "in",
"message": { "update": { "attribute": { "origin": "igp", "med": 200, "local-preference": 100 },
"announce": { "ipv4 unicast": { "101.1.101.1": { "1.1.1.1/32": { "path-information":
"0.0.0.0" } } } } } } } }
```

ExaBGP, pour les programmeurs

https://github.com/Exa-Networks/exabgp

~~SDN~~ Network Automation avec BGP

(et un peu de programmation)



Ouvert (utilisable avec tout langage du bash au C++)

Plus de détails disponibles en suivant ces liens

- http://github.com/Exa-Networks/exabgp/wiki/Related-articles
- http://thomas.mangin.com/data/pdf/

ExaBGP, Configuration

```
$ cat qa/conf/api-announce.conf
process add-remove {
        run ./api-announce.run;
        encoder json;
}

neighbor 127.0.0.1 {
        router-id 1.2.3.4;
        local-address 127.0.0.1;
        local-as 1;
        peer-as 1;
        group-updates false;

        capability {
                graceful-restart;
        }

        api {
                processes [ add-remove ];
        }
}
```

Configuration actuelle (master branch on git)

Pour 3.4 – voir mes anciennes présentations

./api-announce.run ne reçoit que le message de terminaison d'ExaBGP

Un grand nombre d'exemples dans le répertoire:

$ find **$exabgp/qa/conf**

La seule documentation toujours à jour (tristement)

ExaBGP, Configuration avancée

```
api {
        processes [process-1 process-2 ];
        neighbor-changes;
        receive {
                parsed;
                update;
        }
        send {
                packets;
        }
}
```

processes:
Les processus auxquels il faut transmettre les données

receive: paquets reçus du peer
send: paquets envoyés au peer

packets: format binaire (json)
parsed: format humain (json/texte)
consolidate: ajouter le format binaire au format json

Ordre: 1 – binaire, 2– json/texte

Afin de toujours avoir le binaire si necessaire.

```
api {
        processes [ process-3];
        send {
                consolidate;
                open;
        }
}
```

ExaBGP, Programming (en python)

```
$ cat qa/conf/api-announce.run
#!/usr/bin/env python

import sys
import time

messages = [
'announce route 1.1.0.0/24 next-hop 101.1.101.1',
'announce route 1.2.0.0/25 next-hop 101.1.101.1',
]

time.sleep(2)

while messages:
        message = messages.pop(0)
        sys.stdout.write( message + '\n')
        sys.stdout.flush()
        time.sleep(1)

sys.stdin.readline()
```

Simple loop qui imprime sur STDOUT les routes à annoncer

On peut limiter la diffusion a des annoncements

‘neighbor <IP>[,<IP>]+ announce …’

exemple de lecture et écriture

$ cat **$exabgp**/etc/exabgp/processes/read-write.py

Toute aide de documentation appréciée

Jacques a dit: Unit Testing

```
$ ./qa/bin/conversation

0 addpath                       1 api-add-remove
2 api-announcement              3 api-api
4 api-broken-flow               5 api-check
6 api-eor                       7 api-fast
8 api-flow                      A api-multi-neighbor
B api-multisession              C api-nexthop
D api-nexthop-self              E api-notification
F api-open                      G api-reload
H api-teardown                  I api-vpls
J api-vpnv4                     K generic-attribute
L group                         M ipv46routes4family
N ipv46routes6family            O manual-eor
P name                          Q no-asn4
R parity                        S path-information
T skip                          U split
V unknowncap                    W vpn
X watchdog

starting
--------
0 1 2 3 4 5 6 7 8 A B C D E F G H I J K L M N O P Q R S T U V W X

collected
---------
0 1 2 3 4 5 6 7 8 A B C D E F G H I J K L M N O P Q R S - - V W X
```

```
# LISTING=1 ./qa/bin/conversation
0 addpath                       1 aggregator
…
# SERVER=1 ./qa/bin/conversation
# CLIENT=1 ./qa/bin/conversation

# cat qa/conf/aggregator.conf
neighbor 127.0.0.1 {
…
# cat qa/conf/aggregator.sequence
1:raw:FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF:004802:0000
002D400101004002A02020000E1DD00001B1B400304C
0A801FE400504000000644006O0C00708000046E0DB76
E1BD12010040
```

ExaBGP, Utile ?

Visualisation du réseau

Protection DDOS

- RTBH
- Flow Spec

Programmation dynamique

- Remplacer BGP Best Path
- Répartition de flux

Haute disponibilité

- Anycast
- Active / Passive

- Autre …

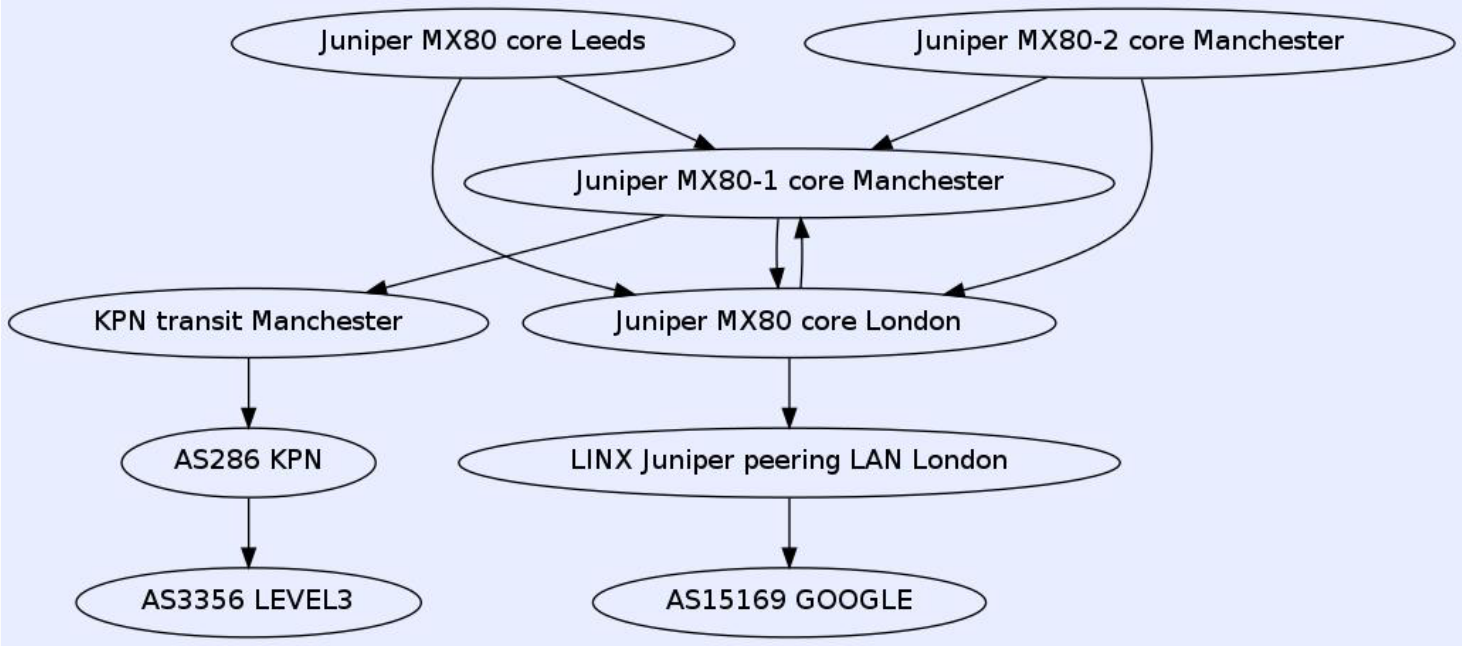
Visualisation

https://github.com/dpiekacz/gixlg

par Daniel Piekacz (874 LOC)

Vision locale

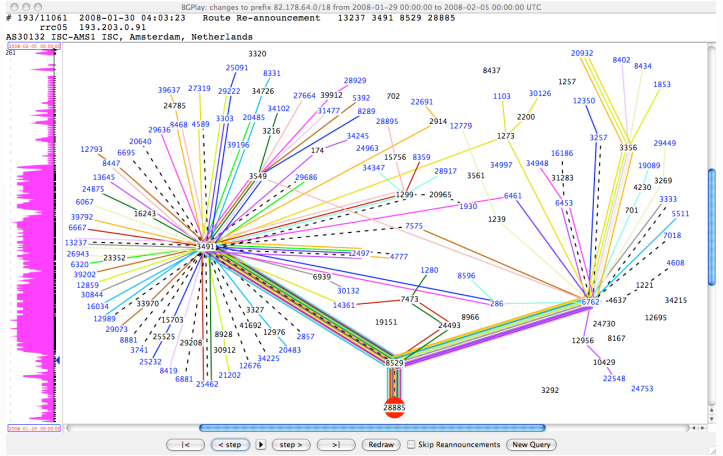
”Hop to Hop”



Visualisation à grande échelle

https://labs.ripe.net/Members/colin_petrie/updates-to-the-ripe-ncc-routing-information-service

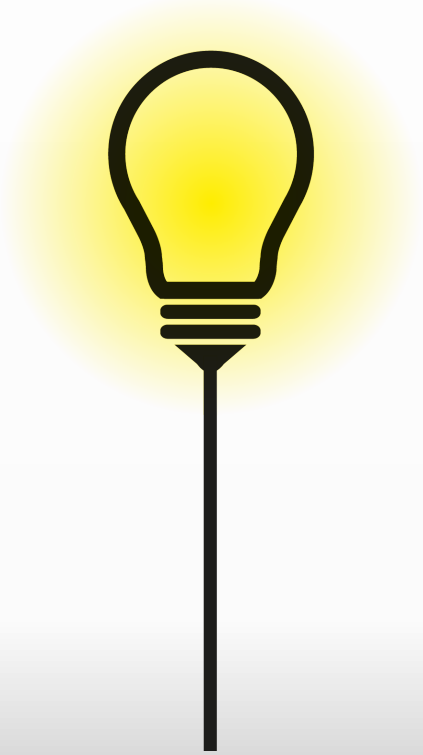
par Colin Petrie



RIPE experimental real-time RIS

- https://tools.ietf.org/html/draft-petrie-grow-mrt-add-paths-00

Programmation dynamique du réseau



A été fait à grande échelle (pas de solution open source cependant)

Offres commerciales similaires

- http://www.noction.com
- http://www.border6.com
- http://www.internap.com/network-services/ip-services/miro-controller/

Répartir le trafic entre plusieurs points d'interconnexion

- 4x1Gb à Manchester, 2x1Gb à London
- Mesurer le trafic par LTS (plusieurs centaines)
- Annoncer chaque /32 indépendamment.

Protéger son réseau

https://github.com/infowolfe/exabgp-edgerouter

Par infowolfe

Bloque automatiquement les listes « Spamhaus DROP/EDROP » et « Emerging Threats » blocklists avec les routeurs d'Ubiquiti (ou autres).

https://www.cymru.com/jtk/misc/utrs.html

par jtk

RTBH collaboratif entre acteurs du net

Protection DDOS

https://github.com/FastVPSEestiOu/fastnetmon

par Pavel Odintsov

Collection de flows (SFLOW, NETFLOW, IPFIX, PCAP, PF_RING, …)
Détecte les anomalies
Injecte des routes IPv4/IPv6/FlowSpec avec ExaBGP (or GoBGP)

Autres sources d'information

- https://www.nanog.org/sites/default/files/tuesday_general_ddos_ryburn_63.16.pdf
- https://www.nanog.org/sites/default/files/tuesday_lt_kristoff_utrs.pdf
- http://perso.nautile.fr/prez/fgabut-flowspec-frnog-final.pdf

Haute disponibilité

http://vincent.bernat.im/en/blog/2013-exabgp-highavailability.html

par Vincent Bernat (534 LOC)

python -m exabgp healthcheck --help

Host HA services

- Annonce un IP de service (/32) par entrée DNS quand le service est fonctionnel
- MED pour la sélection active / passive (EBGP)
- Ou AnyCast vos DNS / NTP / HTTP

❖http://thomas.mangin.com/data/pdf/SYSADMIN 4 - Mangin - BGP for sysadmin.pdf

Solutions similaires

❖https://github.com/pyke369/exabgp-helpers

Haute disponibilité

http://bits.shutterstock.com/2014/05/22/stop-buying-load-balancers-and-start-controlling-your-traffic-flow-with-software/

par Allan Feid

Utilisez vos routeurs comme répartiteur de charge

Load balancing par flow contrôlé par BGP

Demain, ou un peu plus tard ..

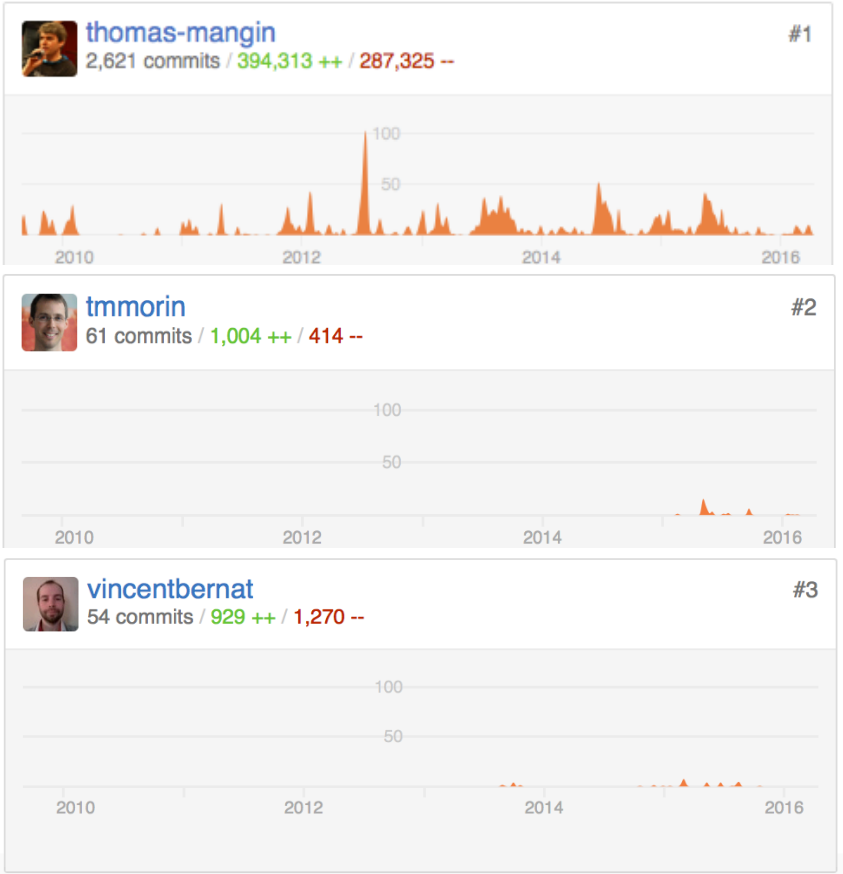
Version 4.0 – cette année (promis peut-être)

- Meilleur format de configuration
- Un CLI .. (enfin !)
- Multi-processus pour décoder les paquets
- Regardez les tickets GitHub pour les promesses

De l'aide serait la bienvenue

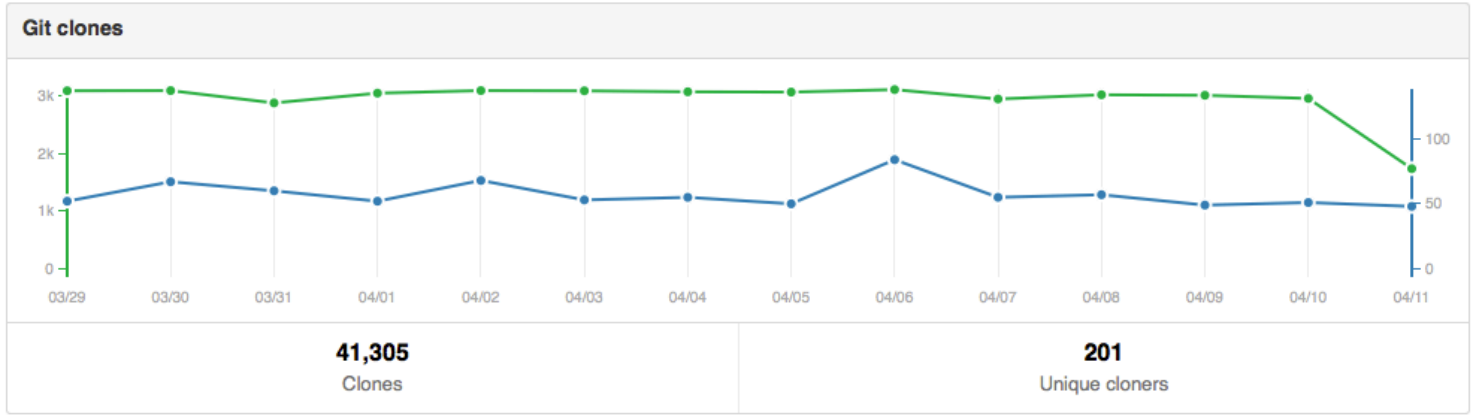
Auteur de plus 95% du code

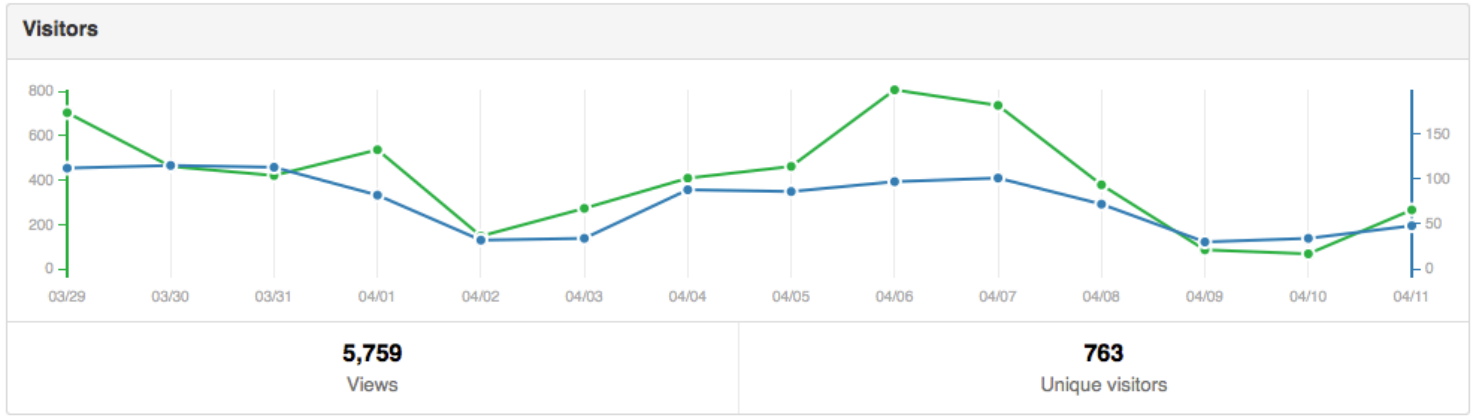
J'aimerais baisser cette statistique ☺



❖https://github.com/Exa-Networks/exabgp/wiki/Other-OSS-BGP-implementations

Note … A propos de ce CI …





50 cloneurs distincts / jour

3,000 clones total / jour

Où est le CI de fou ?

2 clones / minutes

De 30 à 100 visiteurs / jour

De 30 à 700 pages / jour

Finalement: Quelques remerciements

Thomas Morin

Pour le code EVPN

- https://github.com/Orange-OpenSource/bagpipe-bgp

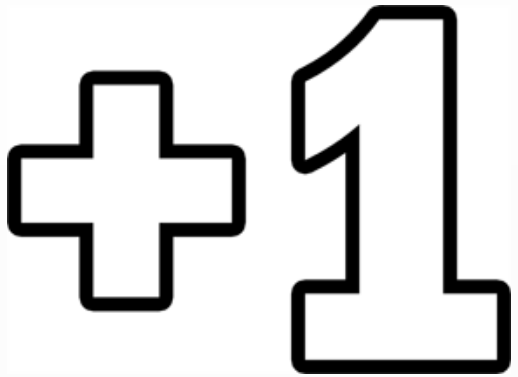
Vincent Bernat

Pour la maintenance Debian

Et l'addition de healthcheck

Pierre-Yves Kerembellec

- https://github.com/pyke369/exabgp-helpers



J'espère n'avoir oublié personne ..

Questions

32 slides plus tard, c'est fini !

J'espère que vous êtes maintenant curieux

Je reste pour le « glou », si vous avez des questions

email: first @ last . com

http://thomas.mangin.com/data/pdf

http://github.com/Exa-Networks/exabgp/wiki/Related-articles

Merci de votre attention

Avez-vous vu notre proxy HTTP « exaproxy » ?