Exploitation du format JSON pour les réseaux et les systèmes

March 21, 2022

1 Un exemple de l'exploitation de fichiers JSON dans le domaine de l'administration des systèmes et des réseaux.

jc est un utilitaire développé en Python (installable via 'pip install jc') qui transforme la sortie de certaines commandes Linux en json (voir ici).

jc est aussi utilisable en tant que package Python: il peut "parser" le résultat d'une commande shell.

L'appel de cette commande se fait via le module subprocess de Python comme on peut le voir ici:

1.1 Exemple d'utilisation du package jc

[ ]:
```
import subprocess
import jc

cmd_output = subprocess.check_output(['dig', 'example.com'],
                                     text=True)
data = jc.parse('dig', cmd_output)
print(data)
```

```
[{'id': 64805, 'opcode': 'QUERY', 'status': 'NOERROR', 'flags': ['qr', 'rd',
'ra'], 'query_num': 1, 'answer_num': 1, 'authority_num': 0, 'additional_num': 1,
'opt_pseudosection': {'edns': {'version': 0, 'flags': [], 'udp': 65494}},
'question': {'name': 'example.com.', 'class': 'IN', 'type': 'A'}, 'answer':
[{'name': 'example.com.', 'class': 'IN', 'type': 'A', 'ttl': 74353, 'data':
'93.184.216.34'}], 'query_time': 4, 'server': '127.0.0.53#53(127.0.0.53)',
'when': 'lun. mars 21 22:20:11 CET 2022', 'rcvd': 56, 'when_epoch': 1647897611,
'when_epoch_utc': None}]
```

1.2 Exploitation des résultats de la commande 'ss -tun'

Utilisez jc et subprocess afin de parser la commande "ss -tun" afin de réaliser des statistiques par état des sockets.

La liste des états des sockets à analyser est la suivante:

'ESTAB','LISTENING','FIN-WAIT-1','FIN-WAIT-2','TIME-WAIT','CLOSED','CLOSE-WAIT','SYNC-SENT','SYNC-RECV','LAST-ACK','CLOSING' Voilà ce qui est attendu comme sortie de votre programme:

```
état: ESTAB nombre d'entrées dans cet état: 42
état: LISTENING nombre d'entrées dans cet état: 0
état: FIN-WAIT-1 nombre d'entrées dans cet état: 0
état: FIN-WAIT-2 nombre d'entrées dans cet état: 0
état: TIME-WAIT nombre d'entrées dans cet état: 0
état: CLOSED nombre d'entrées dans cet état: 0
état: CLOSE-WAIT nombre d'entrées dans cet état: 5
état: SYNC-SENT nombre d'entrées dans cet état: 0
état: SYNC-RECV nombre d'entrées dans cet état: 0
état: LAST-ACK nombre d'entrées dans cet état: 0
état: CLOSING nombre d'entrées dans cet état: 0
Nombre de 'peer_address' : 15
Nombre de 'peer_port' : 15
```

2 Exploitation d'annonces BGP au format JSON avec Python et jq

BGP est le protocole de routage dynamique de l'internet. Les annonces BGP permettent aux routeurs d'ajuster leurs tables de routages.

Une annonce BGP contient un 'AS-PATH' c'est à dire un chemin des AS que les paquets IP doivent traverser afin d'atteindre un préfixe réseau.

Il n'est nul besoin de disposer d'un routeur de coeur de l'internet afin d'accéder aux annonces BGP: Le site du "RIS live" mis en place par le "Ripe NCC" permet de récupérer les annonces BGP que reçoivent les routeurs de l'internet.

2.1 Le service RIS

L' article suivant de Stéphane Bortzmeyer décrit le service "RIS" et propose un programme Python afin de "dumper" au format JSON les informations fournies au travers d'une Web Socket en mode Asynchrone.

Installez le package websockets pour Python3 et amusez-vous à récupérer les informations transmises par RIS durant une minute en utilisant le script Python ris-live.py.

Qu'en déduisez-vous de l'intensité du trafic des annonces BGP ?

Créez un code Python permettant d'affichez les 3 premiers enregistrements du fichier ris.json comme le fait la commande jq suivante:

```
cat ris.json | jq --slurp '.[:3][].data.peer_asn'
"8896"
"39107"
"28824"
```

2.2 Calcul des longueurs moyennes et médianes du "path" bgp

Le "path" BGP représente la liste des "Autononous System" traversés afin d'atteindre un réseau de destination. C'est la métrique centrale du protocole BGP.

Quelle est la longueur moyenne et la longueur médiane des chemins (path) extraits du fichier ris.json ? Il peut y avoir plusieurs annonces par ris messages. Vous éliminerez les chemins de longueurs nulles.

Quelle est le nombre moyen et médian de réseaux annoncés ?

Quel est le pourcentage de réseaux IPV4 et IPV6 parmi les nexthop (éliminez les doublons) ?

Nb: On peut déterminer l'appartenance d'un réseau à IPV4 ou IPV6 grace au package 'builtin' ipaddress et à l'instruction 'isinstance' qui vérifie que l'instance provient bien de sa classe Python:

```
ipv=ip_network(reseau)
if isinstance(ipv,IPv6Network):
   ....
```

Quel est le pourcentage de route provenant d'un IGP ?

2.3 Stockage des données extraites de JSON dans un "dataframe" Pandas

"Pandas" est un package Python pour la lecture, la mise en forme et le traitement de données. Il vous est demandé de l'installer via pip et de l'utiliser afin de créer une table(dataframe) contenant les informations suivantes.

```
       Action              Préfixes AS_origine     ASpath                          lenPath
0      UPDATE       5.133.224.0/20     196925     [8896, 3356, 20771, 196925]        4
1      UPDATE     149.126.112.0/20     196925     [8896, 3356, 20771, 196925]        4
2      UPDATE     149.126.119.0/24     196925     [8896, 3356, 20771, 196925]        4
3      UPDATE     149.126.118.0/24     196925     [8896, 3356, 20771, 196925]        4
4      UPDATE     149.126.117.0/24     196925     [8896, 3356, 20771, 196925]        4
...       ...                  ...        ...
```

Vous pouvez structurer votre code de la façon suivante la suivante:

```
import pandas as pd
pd.set_option('display.expand_frame_repr', False)
listeRow = []

for....
   A vous de jouer

# création du dataframe df

df = pd.DataFrame(listeRow, columns =['Action', 'Préfixes', 'AS_origine','ASpath','lenPath'])

# Affichage
print(df)
```

2.3.1 Grapher les données des 20 premières lignes du dataframe df.

A l'aide du package plot (un wrapper de pyplot dans pandas), de sa méthode scatter, et de la méthode head du dataframe vous afficherez le graphique suivant à partir des données des 20 premières

lignes du dataframe généré précédemment:

[ ]:
```
from IPython.display import Image
Image(filename='images/pandagraph.png')
```

[ ]: