

How NOT to Monitor The Global DFZ

Overview

- Christmas, I've got two weeks off work, loads of wine and food
- I'm feeling "cody"



Overview

- The global BGP DFZ contains a non-trivial amount of “unacceptable” data (in my opinion)
- Some stuff is binary “this is bad”:
 - RPKI invalids
 - full table leaks

Overview

- Some stuff is not so clear.
- What's an acceptable...
 - AS path length?
 - prefix length?
 - number of communities?
 - Number of updates per peer/origin/prefix, per second?

Overview

- Naming and shaming repeat offenders is proven to be less effective when compared to training those offenders
 - ...but significantly more entertaining
- We have a shortage of mavericks which nobody asked for
- Did I mention the wine?

Overview

Epiphany: what the world needs is a 'Top Trumps' of worst DFZ operators

Overview

- Why wouldn't this be great?!?!





DFZ Name and Shame

451 Tweets



Follow

DFZ Name and Shame

@bgp_shamer

I'm a bot who tweets once per day, naming the ASNs who have sent garbage into the [#BGP](#) [#DFZ](#) in the past 24 hours. Written by [@jwbensley](#). Stop it you shits!

In a docker container Joined January 2022

0 Following 105 Followers

← Thread



DFZ Name and Shame

@bgp_shamer



Thread for 2022/04/12. Full details at github.com/DFZ-Name-and-S...

7:09 am · 13 Apr 2022 · DFZ Name and Shame



DFZ Name and Shame @bgp_shamer · 13 Apr



Replying to @bgp_shamer

For 2022/04/12 27116588 BGP UPDATES were parsed. 26297839 UPDATES contained prefix advertisements. 2211059 UPDATES contained prefix withdraws.



DFZ Name and Shame @bgp_shamer · 13 Apr



Replying to @bgp_shamer

Bogon prefixes with most origin ASNs per prefix: 1 bogon prefix(es) had 2 origin ASNs.



DFZ Name and Shame @bgp_shamer · 13 Apr



Replying to @bgp_shamer

Longest AS path: 1 prefix(es) had an AS path length of 255 ASNs.



DFZ Name and Shame @bgp_shamer · 13 Apr



Replying to @bgp_shamer

Longest community set: 69 prefix(es) had a community set length of 442 communities.



main | [dnas_stats](#) / [2022](#) / [04](#) / [12](#) / [20220412.txt](#) Go to file ...

dnasbot Adding report(s) for 20220412 Latest commit 4c5ab5a yesterday [History](#)

0 contributors

103 lines (91 sloc) | 250 KB Raw Blame

```
1 For 2022/04/12 27116588 BGP UPDATES were parsed. 26297839 UPDATES contained prefix advertisements. 2211059 UPDATES contained
2
3 Bogon prefixes with most origin ASNs per prefix: 1 bogon prefix(es) had 2 origin ASNs.
4 Prefix 192.88.99.0/24 from origin ASN(s) AS36732 (COMCAST-36732) AS6939 (HURRICANE)
5
6 Longest AS path: 1 prefix(es) had an AS path length of 255 ASNs.
7 Prefix 91.246.12.0/24 via peer AS137409 (GSLNETWORKS-AS-AP) from origin ASN(s) AS51196 (GOLDTELECOM). AS Path length 255: AS1
8
9 Longest community set: 69 prefix(es) had a community set length of 442 communities.
10 Prefix 212.193.166.0/24 via peer AS34288 (AS34288) from origin ASN(s) AS60519 (SHLH-AS). Community set length 442: 0:2854 0:3
11 Prefix 212.193.166.0/24 via peer AS47787 (EDGOO) from origin ASN(s) AS60519 (SHLH-AS). Community set length 442: 0:2854 0:322
12 Prefix 80.92.164.0/22 via peer AS47787 (EDGOO) from origin ASN(s) AS60921 (FastRu). Community set length 442: 0:2854 0:3226 0
13 Prefix 80.92.164.0/24 via peer AS47787 (EDGOO) from origin ASN(s) AS60921 (FastRu). Community set length 442: 0:2854 0:3226 0
```

Overview

- It's a WIP, but I've already learned a lot...
 - You need a DFZ data source
 - You need code to process the DFZ data
 - You need compute/storage to run code and store bytes

Data

Data

- Setup a BGP peering and parse the BGP UPDATES
 - I'm too lazy to write a new parser
 - Existing libraries are poop
 - Realtime data is not resilient (self hosting / shoestring)
- Setup a BGP peering and use a JSON exporter
 - Realtime data
- RIS Live “fire-hose” API: <https://ris-live.ripe.net/>
 - Realtime data
- *** Use MRT dumps ***
 - Parser libraries exist, asynchronous data consumption

Data

- RFC6396 “MRTs” - Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format
- Encode routing data in a binary file using a TLV structure
 - Note: turns out, not parser friendly
- Mainly used for BGP, also supports OSPFv2, OSPFv3, ISIS

Data

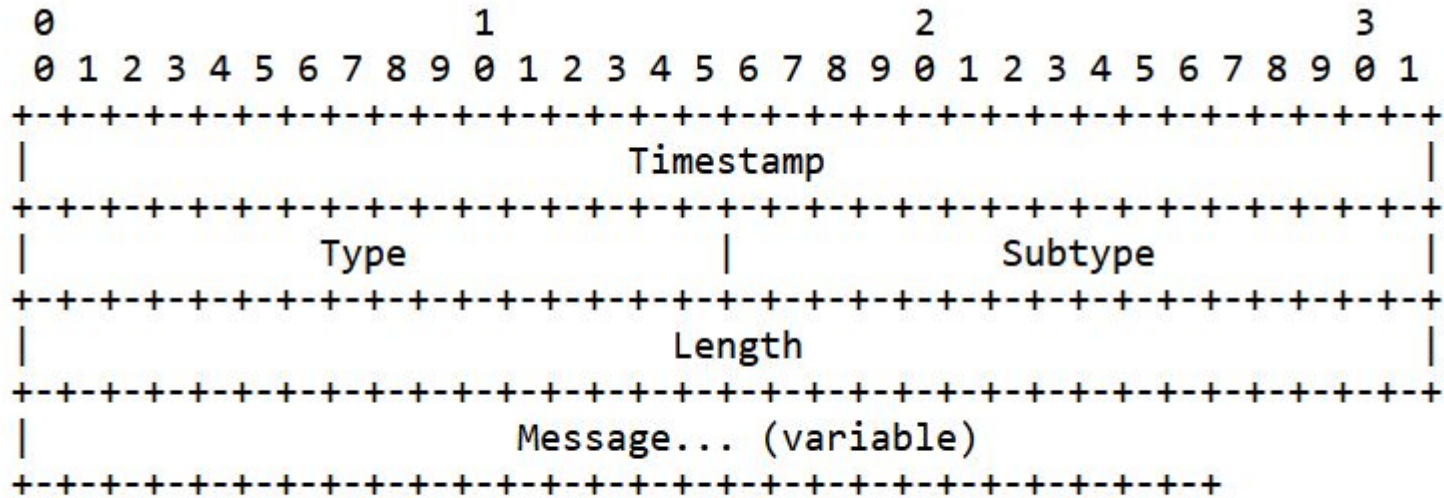


Figure 1: MRT Common Header

Data

- Type: TABLE_DUMP_V2
- Subtypes:
 - PEER_INDEX_TABLE
 - RIB_IPV[4|6]_[UNICAST|MULTICAST]
 - RIB_GENERIC

Data

- Type: BGP4MP_ET
- Subtypes:
 - BGP4MP_STATE_CHANGE
 - *** BGP4MP_MESSAGE ***
 - *** BGP4MP_MESSAGE_AS4 ***
 - BGP4MP_STATE_CHANGE_AS4
 - BGP4MP_MESSAGE_LOCAL
 - BGP4MP_MESSAGE_AS4_LOCAL

Data

- Public MRT archives in descending order of shitness:
 - PCH [Raw Routing Data](#) - total shite
 - RouteViews [MRT Archive](#) - middle shite
 - RIPE RIS [Raw Data](#) - a “little bit” shit

Data

- Note: these MRT archives are all from IXPs



Data

- Turns out - MRT files contain a lot of poop
 - All message types (BGP OPEN, BGP KEEPALIVE etc.)
 - BGP state changes
 - Empty BGP UPDATES
 - Some UPDATES have no list of withdrawn routes
 - Some UPDATES have an empty list of withdrawn routes
 - BGP is IPv4 biased -> v6 NLRI's are MP_REACH_NLRI's

Data

- In summary:
 - Public MRT archives are “OK” but they are all IXPs, no Tier 1 / Tier 2 transits
 - MRTs contain their own poop
 - Poor availability of MRT parsing libraries (<https://bgpkit.com/> is coming to the rescue!)

Code

Code

- Short term: I thought, I “only” need to parse a few MRT files, generate some basic stats, and then Tweet about them
- Long term:
 - Tag offending networks in Tweets
 - Send email to peeringDB contacts “yoo is teh b0g0n”

Code

- Why not write this in Python?
 - [mrtparse module](#)
 - [tweepy module](#)
 - [Github module](#)
 - [PeeringDB module](#)
 - I already mentioned I'm lazy

Code

- DO NOT write high performance code in Python



Code

- Decompressing and parsing a “large” MRT file (~150MB RIB dump) into a Python object uses a lot of memory (circa 1GB)
- Single threaded parsing is too slow
- Python multithreading is shite
- Python GIL means multi-processing requires full memory copy
 - 8 cores/proc's == 8GB of memory to parse 150MB file!

Code

- Split MRT files into *\$number_of_cpu* files
- Splitting is single threaded
- Parse each chunk on a separate CPU core
- Merging of parsed results is single threaded
- MRT files have no total length, no index, they are pure TLVs

Code

- Using PyPy3 to resolve the splitting/merging performance

Code

- Python is shit in other ways too
 - Dynamically typed (using mypy)
 - Same code has difference results in venv or pypy
 - Requires extensive unit testing

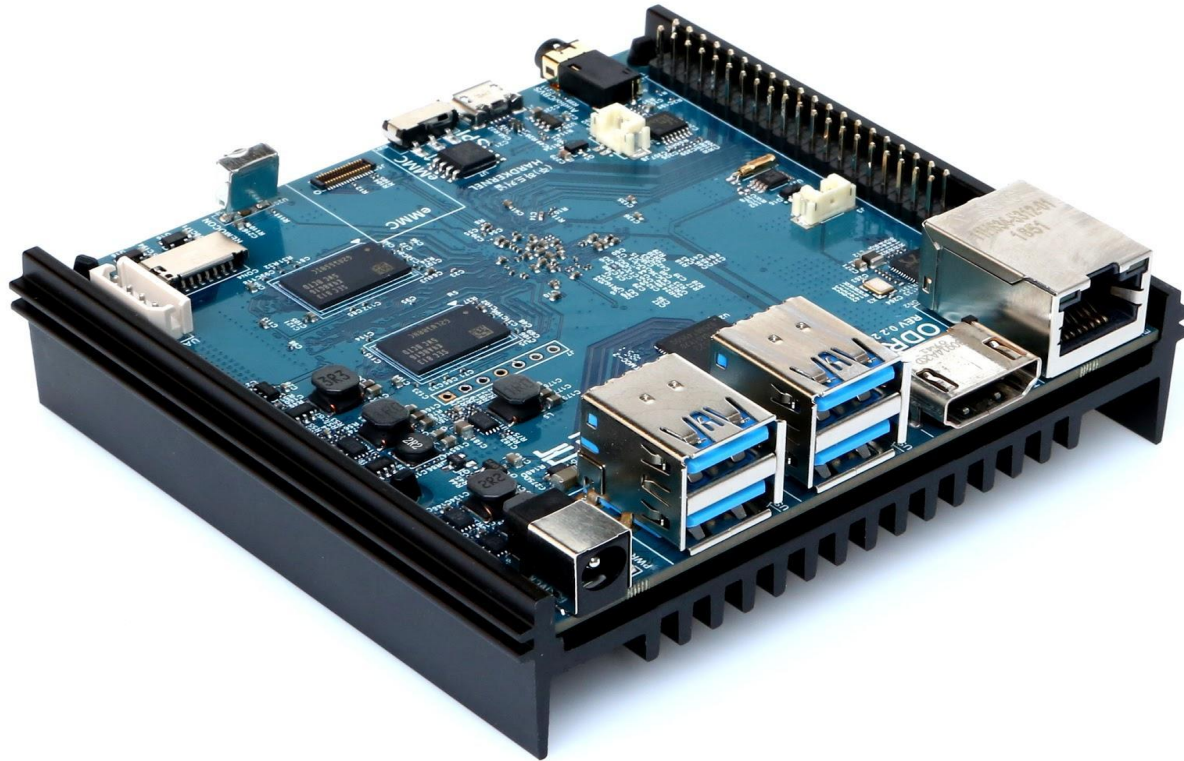
Compute

Compute

- Where will you store the raw data?
- Where will your code run?
- Where will you store the results?

Compute & Storage

- DON'T use an SBC you had lying around



n2_shat.jpg

Compute & Storage

- ODROID-N2
 - Circa 80 EUROS
 - Quad-core ARM Cortex-A73 @ 1.8Ghz
 - Dual-core ARM Cortex-A53 @ 1.9Ghz
 - 4GM RAM (–250MBs for GPU)
 - 16GB eMMC for OS + Redis DB
 - 64GB USB stick for “mass storage”

Compute & Storage

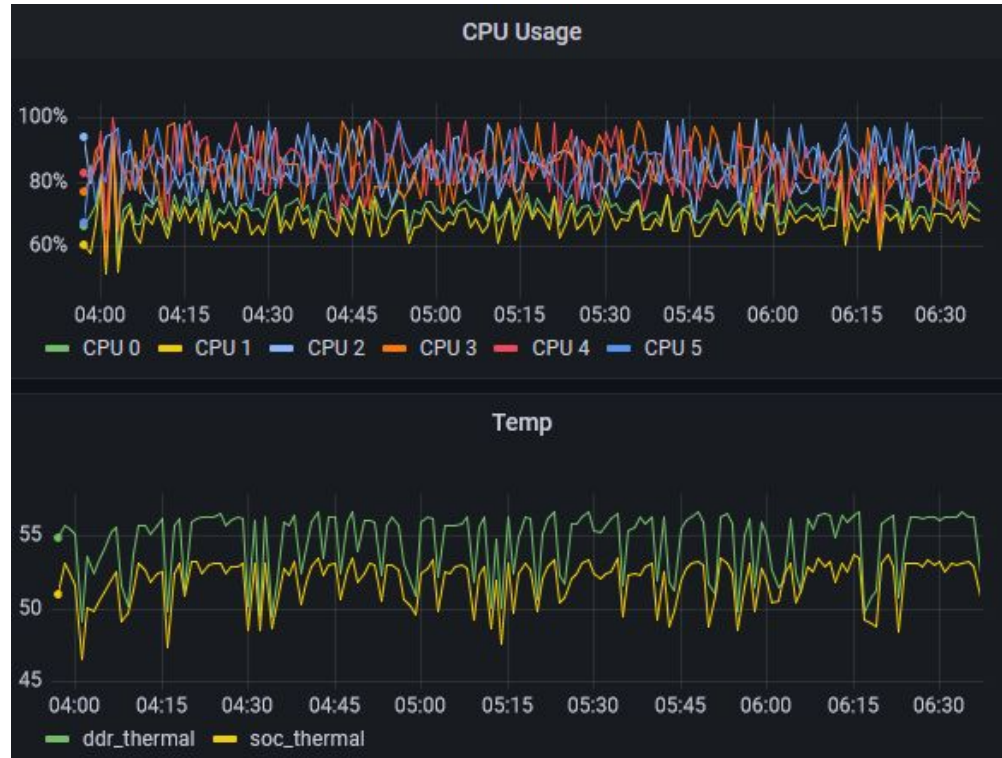
- Parsing in Python is killing the polar bears

```
1 [|||||100.0%] 4 [|||||100.0%]
2 [|||||100.0%] 5 [|||||100.0%]
3 [|||||100.0%] 6 [|||||100.0%]
Mem[|||||1.48G/3.63G] Tasks: 69, 364 thr; 6 running
Swp[|||||0K/0K] Load average: 5.52 5.40 4.31
Uptime: 8 days, 03:38:58
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
88758	root	20	0	345M	111M	18148	R	99.6	3.0	0:38.24	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove
88757	root	20	0	351M	117M	18148	R	98.5	3.2	0:38.07	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove
88759	root	20	0	345M	110M	18164	R	98.0	3.0	0:38.38	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove
88760	root	20	0	348M	114M	18156	R	98.0	3.1	0:37.98	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove
88761	root	20	0	350M	116M	18156	R	97.4	3.1	0:37.98	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove
88762	root	20	0	350M	116M	18156	R	96.3	3.1	0:37.92	/opt/pypy3.8-v7.3.7-aarch64/bin/pypy3 /opt/dnas/dnas/scripts/parse_mrts.py --yesterday --update --remove

Compute & Storage

- Parsing in Python is killing the polar bears



Compute & Storage

- It's also killing my patients

```
[13918.786416] [ 6107]    0 6107   85757   14949    66     4     0     0 pypy3
[13918.786418] [ 6157]    0 6157  132270  117368   257     3     0     0 pypy3
[13918.786421] [ 6158]    0 6158  150237  134938   292     3     0     0 pypy3
[13918.786423] [ 6159]    0 6159  131028  116025   253     3     0     0 pypy3
[13918.786425] [ 6160]    0 6160  136105  120350   263     3     0     0 pypy3
[13918.786428] [ 6161]    0 6161  147038  132050   286     3     0     0 pypy3
[13918.786430] [ 6162]    0 6162  128653  113863   248     3     0     0 pypy3
[13918.786432] [ 6172]  1000 6172    1436     312     8     3     0     0 htop
[13918.786435] Out of memory: Kill process 6158 (pypy3) score 142 or sacrifice child
[13918.788521] Killed process 6158 (pypy3) total-vm:600948kB, anon-rss:539744kB, file-rss:0kB, shmem-rss:8kB
```

Compute & Storage

- Currently 912 MRT files are parsed per day
- ~27M BGP UPDATES
- About 8 hours to download and parse the stats for a day
because I made a pipeline using docker-compose, which is currently broken :(

Lessons Learned

Lessons Learned

- Python shit -> use Go or Rust
- MRTs shit -> Use RIS-Live or equivalent peerings
- SBCs are not “compute” -> Memory and storage bandwidth

Lessons Learned

- Haters gunna hate


← Tweet



James Bensley
@jwbensley

...

I have created a Twitter bot which Tweets once a day about the utter shite polluting the [#BGP](#) [#DFZ](#) during the previous 24 hours - give it a follow and a retweet to raise awareness of all the operators increasing my edge CPU my memory requirements!

 **DFZ Name and Shame** @bgp_shamer · Mar 22
Longest community set on 2022/01/11: 108 prefix(es) had a comm set length of 194 communities
[Show this thread](#)

8:34 PM · Mar 22, 2022 · Twitter Web App

← Tweet



DFZ Name and Shame @bgp_shamer · Mar 22
Longest community set on 2022/01/11: 108 prefix(es) had a comm set length of 194 communities

...

23 1 7



Nat Morris
@natmorris

...

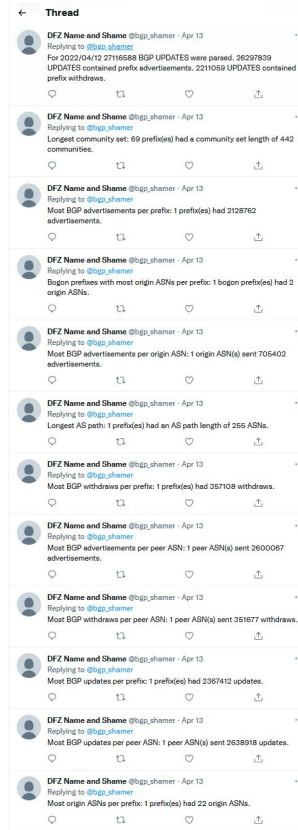
Replying to @bgp_shamer

[@dtemkin](#) think this is one of yours

9:50 PM · Mar 22, 2022 · Twitter for iPhone

Lessons Learned

- A Twitter thread is not the place to distribute a lot of information



End

Bot: https://twitter.com/bgp_shamer

Daily reports: https://github.com/DFZ-Name-and-Shame/dnas_stats