nftlb intro

nftables load balancer

Netfilter Workshop, June 2018

nftlb intro

NFWS2018

user space daemon that manages nftables rules

https://github.com/zevenet/nftlb

v0.2

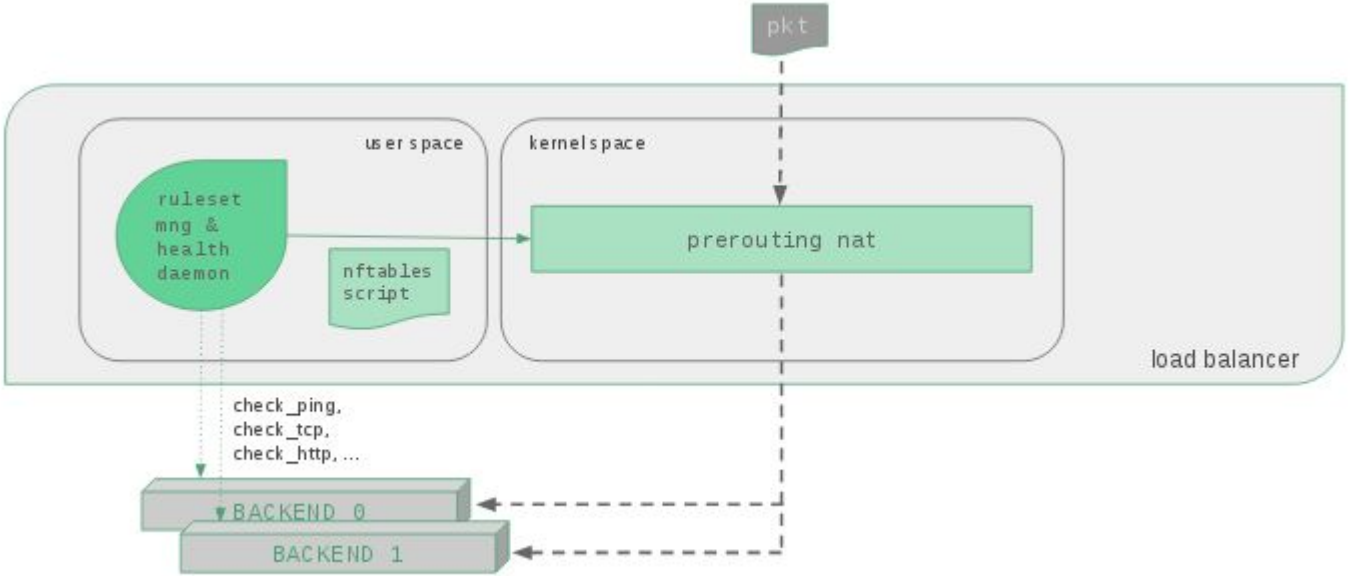
nftlb intro

NFWS2018

First design presented @Netdev 1.1

Development of basic expressions & Benchmarks @Netdev 1.2

nftlb intro


pkt
user space
kernelspace
ruleset
mng &
health
daemon
nftables
script
prerouting nat
load balancer
check_ping,
check_tcp,
check_http, ...
BACKEND 0
BACKEND 1

nftlb intro

Current approaches

LVS

duplicated infrastructure with netfilter, checkers at kernel space, not ready for certain use cases like transparent proxy, multiport, etc. sNAT, TUN and DSR, but not dNAT.

iptables

sNAT, sNAT but not DSR, different binaries for different stacks, too much iptables rules (at least ~2/backend), sequential rules processing and locking problems.

Ex: Kube-proxy has to support both of them

nftlb intro

nft as data path

- Build 3 main topologies used sNAT, dNAT and DSR
- Multiport and multiprotocol handling natively
- Support of both IPv4 and IPv6 traffic seamlessly
- Interact with just one binary/infrastructure
- The virtual services are indexed with vmaps, no need for sequential processing
- Already integrated RCU subsystem
- Improved performance in the most use cases
- Schedulers: weight, round robin, hash and symmetric hash

nftlb intro

… and then, **nftlb** as the control path

- User space daemon
- Optimizes the number of rules and isolation through chains per service
- Multiple virtual services (or farms) support
- Priority support per backend
- Live management via JSON API and http socket
- Web service authentication with a security Key
- Automated testbed

nftlb intro

nft load balancing development

- Kernel space side: easier part, new expressions for traffic distribution
- nft User space side: hardest part, by including new syntax
- nftlb service: simple design and development

```
/
    src/
        main.c
        events.c
        config.c
        model.c
        network.c
        server.c
        nft.c
    tests/
```

nftlb intro


```
Usage: nftlb
  [ -h | --help ]                          Show this help
  [ -l <LEVEL> | --log <LEVEL> ]           Set the syslog level
  [ -c <FILE> | --config <FILE> ]          Launch with the given configuration file
  [ -k <KEY> | --key <KEY> ]           Set the authentication key, otherwise
                                           it'll be generated
  [ -e | --exit ]                          Don't execute the server
  [ -6 | --ipv6 ]                          Enable IPv6 listening port
  [ -H <HOST> | --host <HOST> ]            Set the host for the listening port
  [ -P <PORT> | --port <PORT> ]            Set the port for the listening port
```

nftlb intro


```
{
        "farms" : [
            { <object virtual service 1> },
            { <object virtual service 2> },
            { ... }
        ]
}
```

nftlb intro


```
{
     "name" : "<string>",
     "family": "<ipv4 | ipv6 | dual>",
     "virtual-addr": "<ip address>",
     "virtual-ports": "<port list>",
     "mode": "<snat | dnat | dsr>",
     "protocol": "<tcp | udp | sctp | all>",
     "scheduler": "<weight | rr | hash | symhash>",
     "priority": "<number>",
     "state": "<up | down | off>",
     "backends" : [
          {<object backend 1>},
          {<object backend 2>},
          {...}
     ]
}
```

nftlb intro


```
{
    "name" : "<string>",
    "ip-addr": "<ip address>",
    "weight": "<number>",
    "priority": "<number>",
    "state": "<up | down | off>",
}
```

nftlb intro


```
{    "farms" : [ {
                "name" : "vs01",
                "family" : "ipv4",
                "virtual-addr" : "192.168.0.100",
                "mode" : "snat",
                "protocol" : "80",
                "scheduler" : "hash",
                "priority" : "1",
                "state" : "up",
                "backends" : [ {
                                "name" : "bck0",
                                "ip-addr" : "192.168.1.10",
                                "weight" : "1",
                                "priority" : "1",
                                "state" : "up"
                        },
                        {
                                "name" : "bck1",
                                "ip-addr" : "192.168.1.11",
                                "weight" : "1",
                                "priority" : "1",
                                "state" : "up"
                        } ]
                } ]
}
```

nftlb intro


```
table ip nftlb {
    map tcp-services {
        type ipv4_addr . inet_service : verdict
        elements = { 192.168.0.100 . http : goto vs01 }
    }

    chain prerouting {
        type nat hook prerouting priority 0; policy accept;
        ip daddr . tcp dport vmap @tcp-services
    }

    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
    }

    chain vs01 {
        dnat to jhash ip saddr mod 2 map { 0 : 192.168.1.10, 1 : 192.168.1.11 }
    }
}
```

nftlb intro


```
table ip nftlb {
    map tcp-services {
        type ipv4_addr . inet_service : verdict
        elements = {  192.168.0.100 . http : goto vs01,
                      192.168.0.102 . https : goto vs02   }
    }

    chain prerouting {
        type nat hook prerouting priority 0; policy accept;
        ip daddr . tcp dport vmap @tcp-services
    }

    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
    }

    chain vs01 {
        dnat to jhash ip saddr mod 2 map { 0 : 192.168.1.10, 1 : 192.168.1.11 }
    }

    chain vs02 {
        dnat to numgen inc ip saddr mod 3 map { 0 : 192.168.0.50, 1 : 192.168.0.51, 2 : 192.168.0.52 }
    }
}
```

nftlb intro


```
Virtual service listing.

curl -H "Key: <MYKEY>" http://<NFTLB IP>:5555/farms

Setup a new virtual service.

curl -H "Key: <MYKEY>" -X POST http://<NFTLB IP>:5555/farms -d "@tests/008_snat_ipv4_all_rr.json"

Delete a virtual service.

curl -H "Key: <MYKEY>" -X DELETE http://<NFTLB IP>:5555/farms/lb01

Delete a backend of a virtual service.

curl -H "Key: <MYKEY>" -X DELETE http://<NFTLB IP>:5555/farms/lb01/backends/bck1
```

nftlb intro

Performance figures:

Test Case	Latency	Performance
iptables DNAT	491.36us	256,864 http req/s per core
iptables SNAT	317.08us	262,089 http req/s per core
nft DNAT	312.74us	608,941 http req/s per core
nft SNAT	292.08us	560,976 http req/s per core
nft DSR	374.27us	7,302,517 http req/s per core

nftlb intro

Performance figures:

Test Case	Latency	Performance
iptables DNAT	491.36us	256,864 http req/s per core
iptables SNAT	317.08us	262,089 http req/s per core
nft DNAT	312.74us	608,941 http req/s per core
nft SNAT	292.08us	560,976 http req/s per core
nft DSR	374.27us	7,302,517 http req/s per core

with retpoline enabled:

iptables: 40.77% penalty
nftables: 17.27% penalty

nftlb intro

work to do:

- New expression for Jump Consistent Hash
- Add infrastructure to provide persistence
- Stateless NAT
- Helpers
- Conntrack offload
- and more!