Zone Transfer Performance

Bill Snow, Digicert UltraDNS

2025-02-06

What about it?

- testing zone transfer performance is complicated
- there are data distribution choices

- how do we move data around?
- how do we test for capacity?

Motivation

Managed DNS and

Large installations

ULTRADNS®

Alternatives

- source of truth might be a webapp db, or external primaries
- database replication is a good choice, except
  - head of line problem
  - monolithic scaling

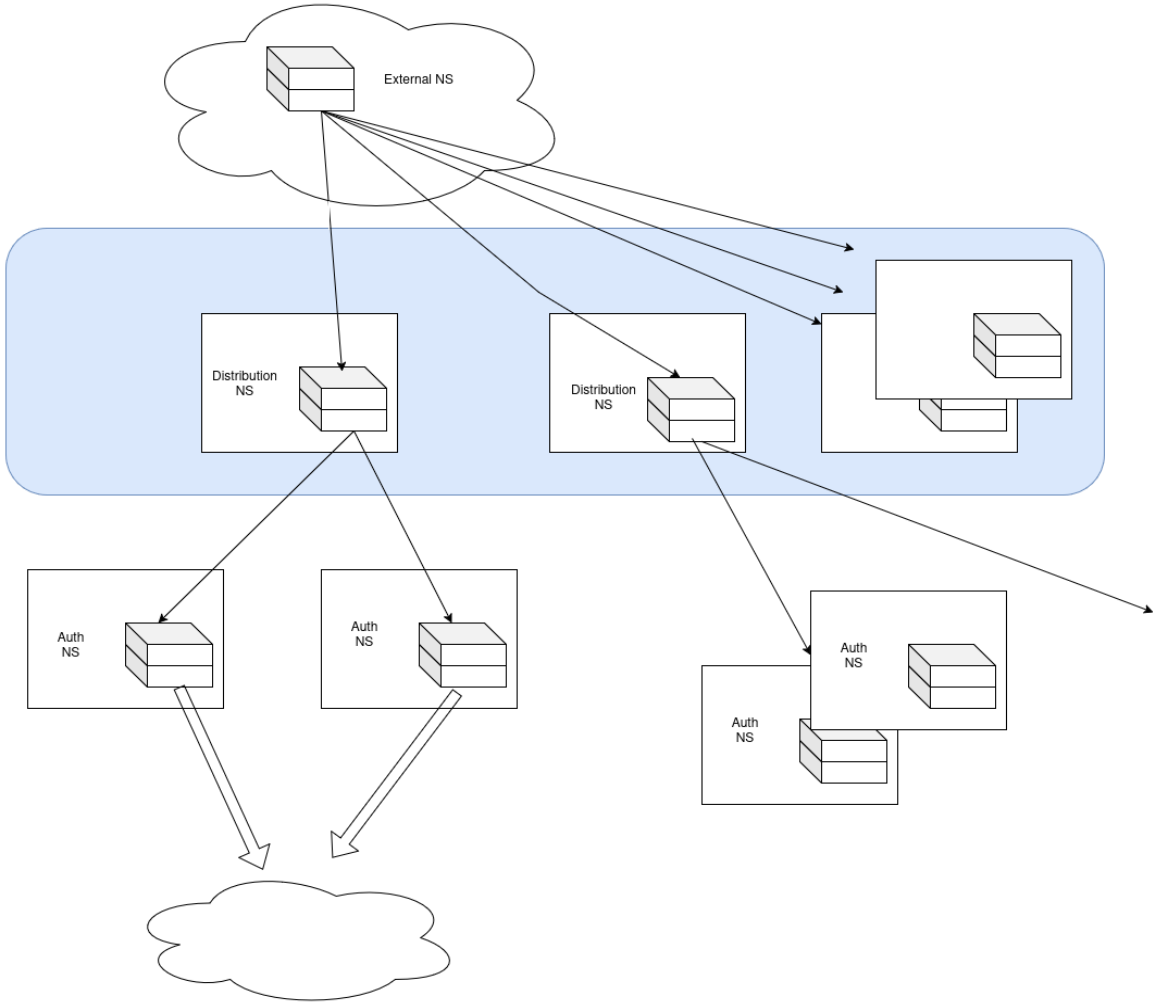
- maybe consider zone transfers

Multi-tiered Installations

- multi-tiered propagation system

- TLDs
- distributor servers transfer in+out

- global distribution in ~5s



Methodology

How are we going to test capacity

Capacity testing

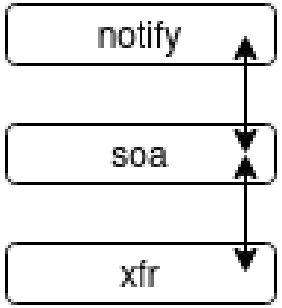
- how do you make a lot of zone transfers at once?
- the **state machine** is annoying
  - notify, soa, transfer messages – what to measure?
  - satisfy ourselves with cold start time - it's not a bad upper bound
  - someone else must do this…

Concurrent sessions

- how do you make a lot of zone transfers at once?
- the **state machine** is annoying
  - notify, soa, transfer messages – what to measure?
  - satisfy ourselves with cold start time - it's not a bad upper bound
  - someone else must do this

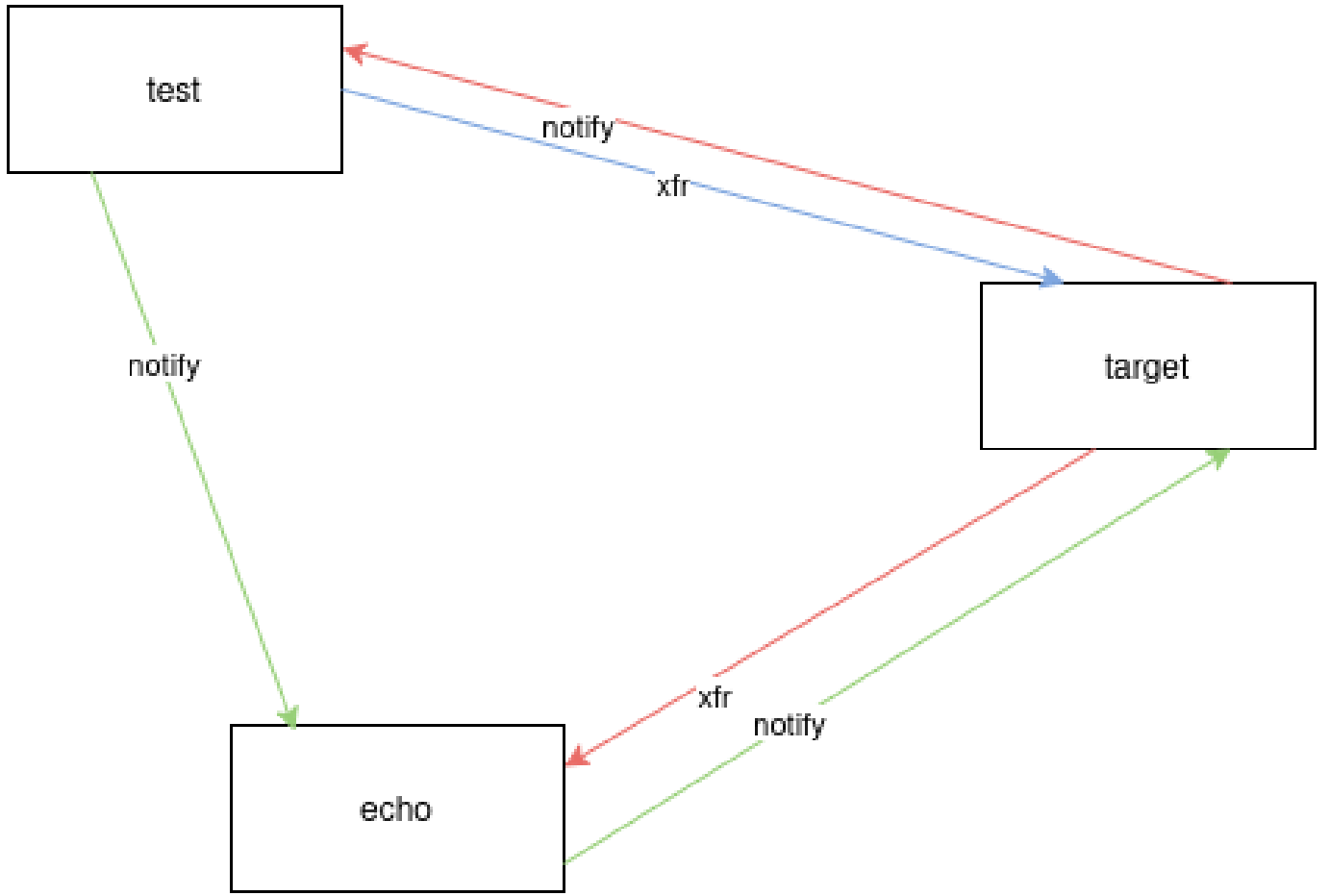
- the webapp people have lots of state machines

```
ab -c <concurrency>
```



Test setup

- just send a bunch of notifies
- how many transfers come back?
- how long do the transfers take?



ULTRADNS®

Echo server

- not exactly an echo server

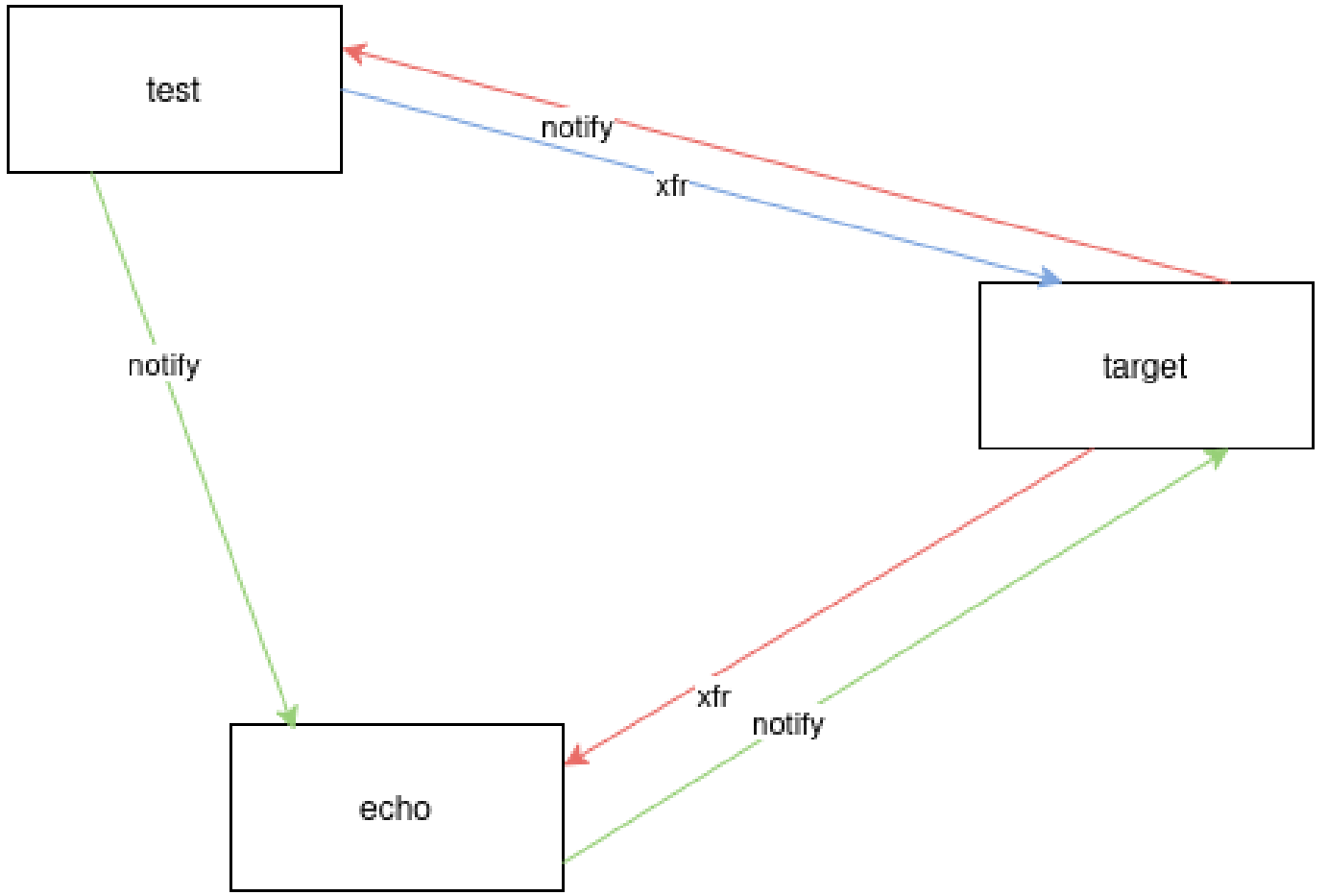
- answers every AXFR request with the same 30 record template
- …and IXFR
- also-notifies
- soa serial number is always the current time in ms
- quick enough. Still hard to make it as quick as knot!
- use long refresh times

- test generates notify load on the echo server
- *every soa request is out-of-date*

What's on the target host?

- bind
- knot
- nsd
- powerdns with postgresql

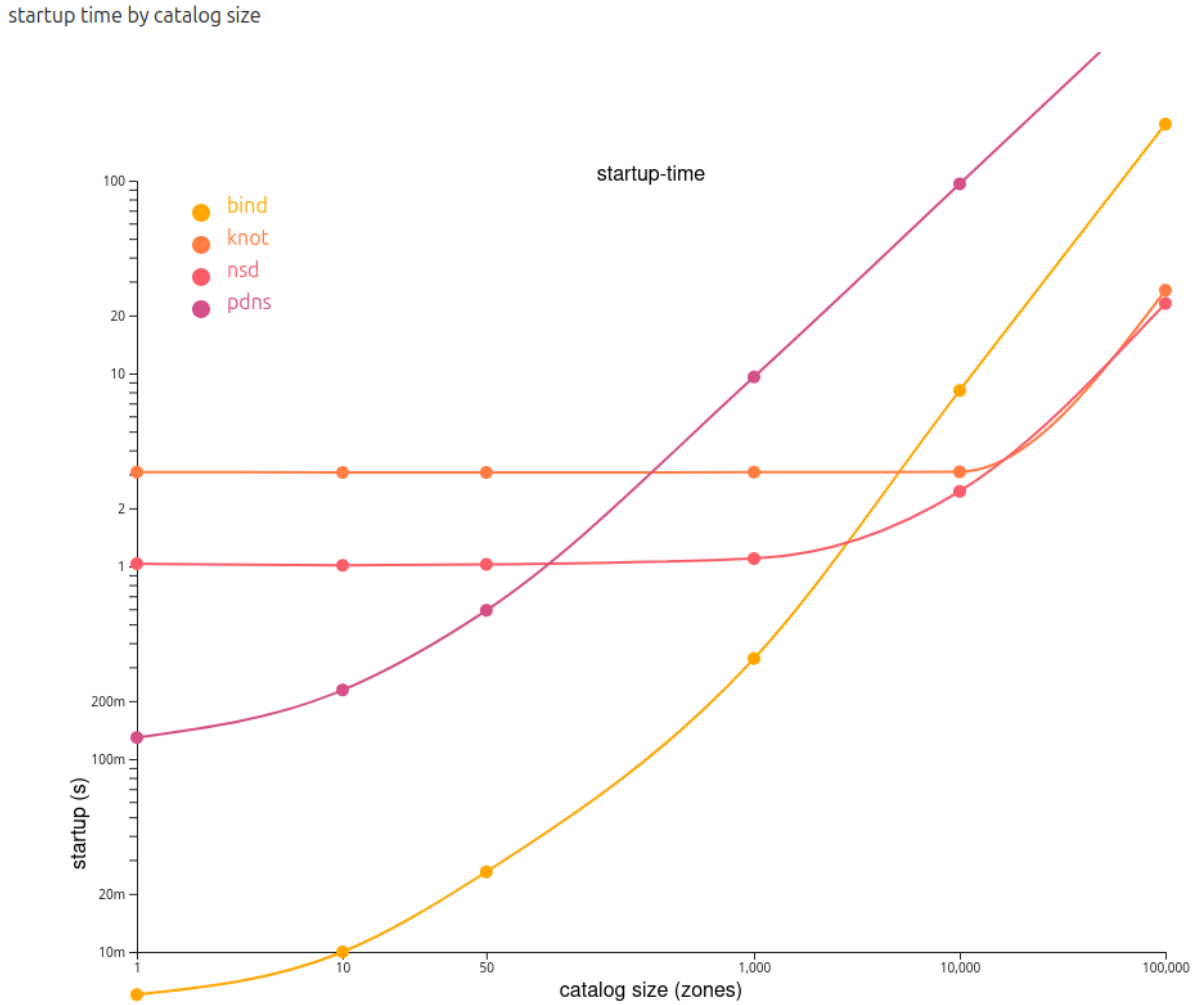
- catalog zones



Benchmark start times

- they're all pretty fast up to 10k
- knot & nsd unbothered for a while
- test setup seems to work

- sanity – ✓

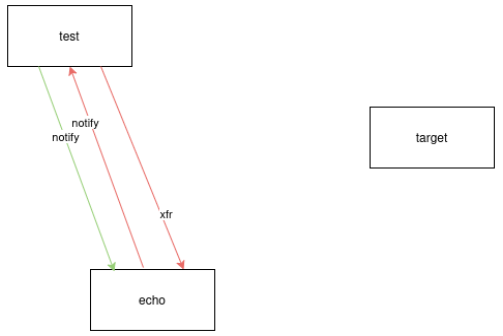


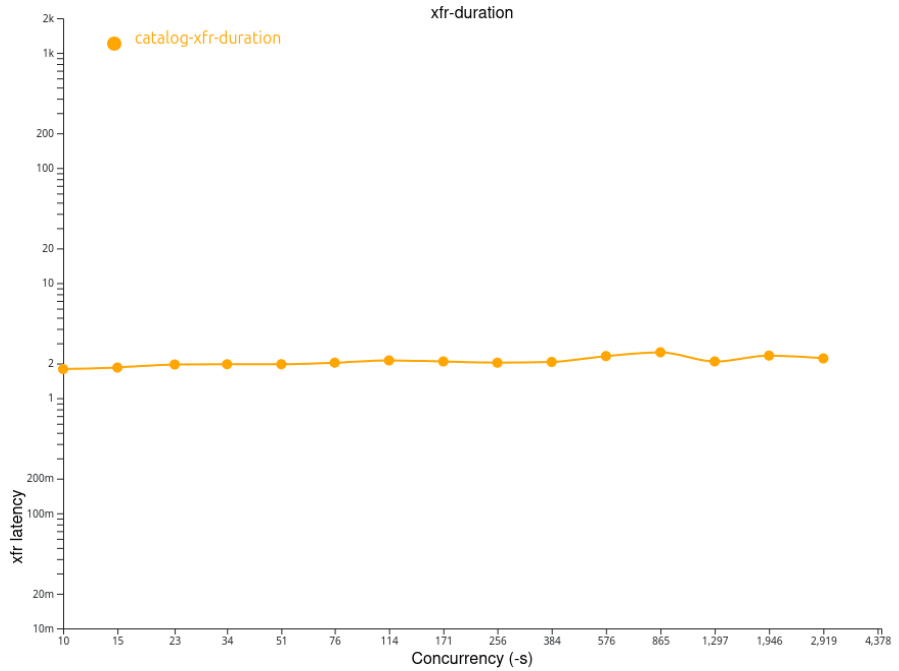
ULTRADNS®

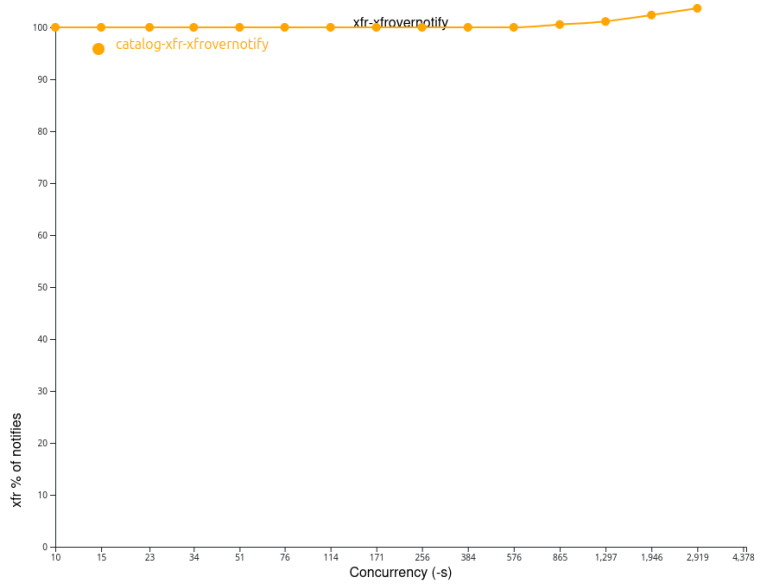
Benchmark the echo server

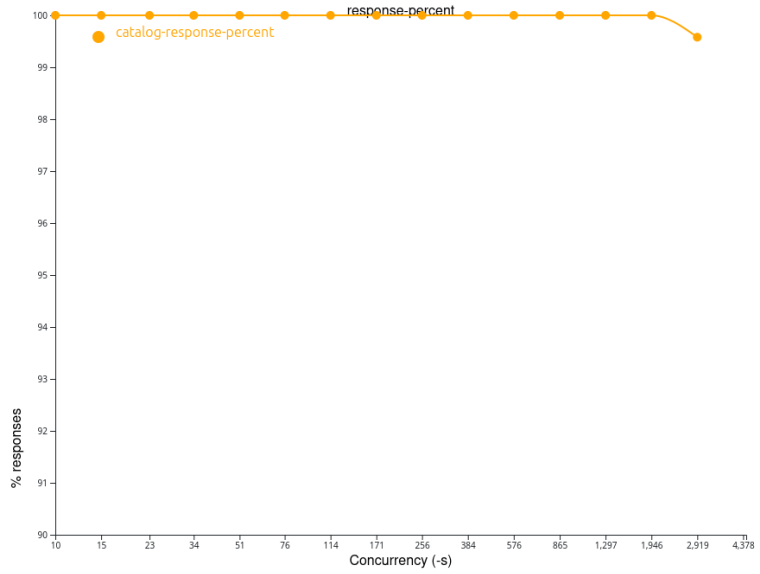
- around ~3k notifies/s
- everything is normal up to 576
- think about that as xfr latency (time to respond)

- sanity – ✓







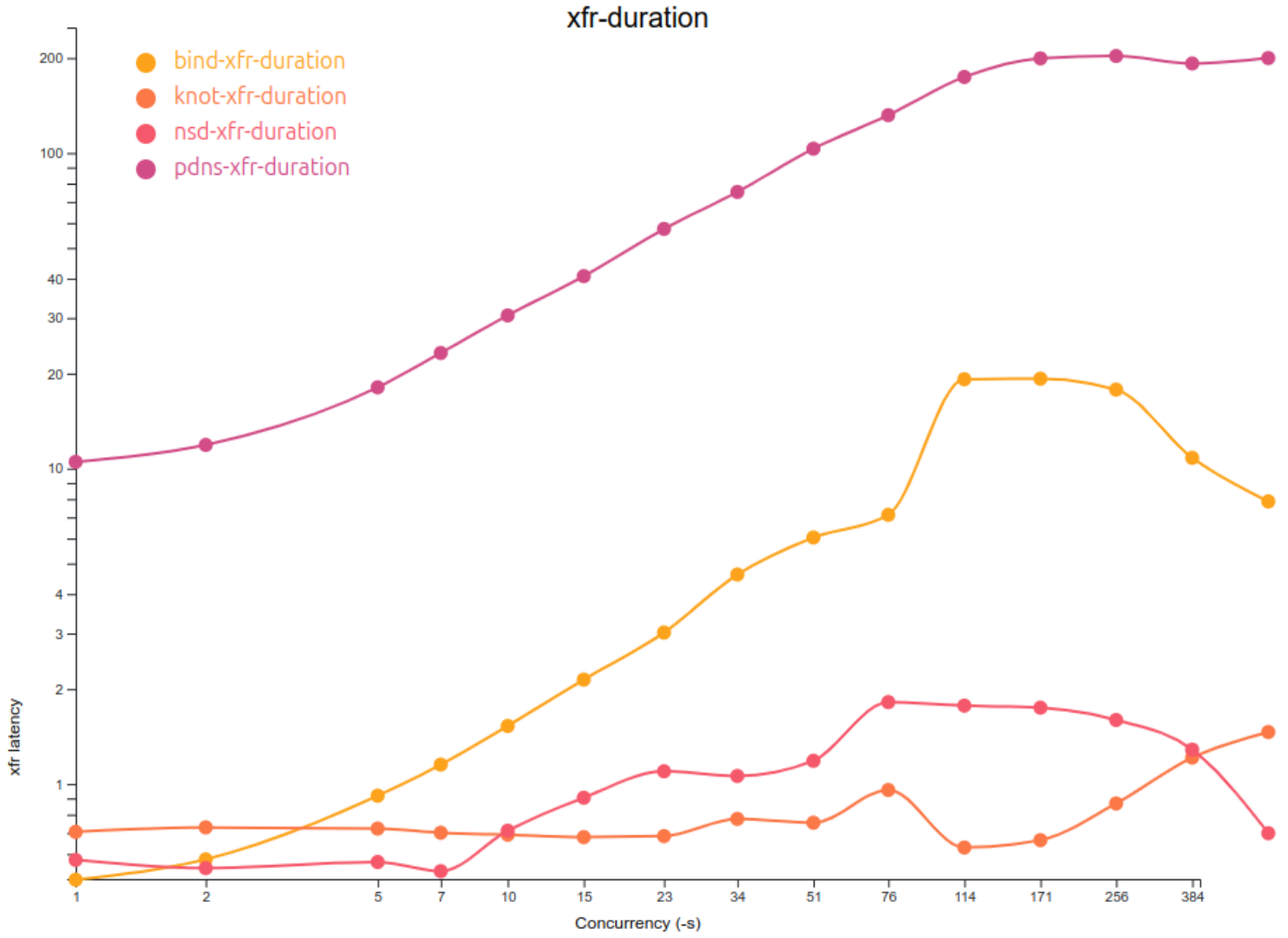


ULTRADNS®

Finally some test results

- total 3k notifies
- pretty fast
- concurrent sessions
- qps <1000 (more later)

- but this is the response time of the target to the xfr request

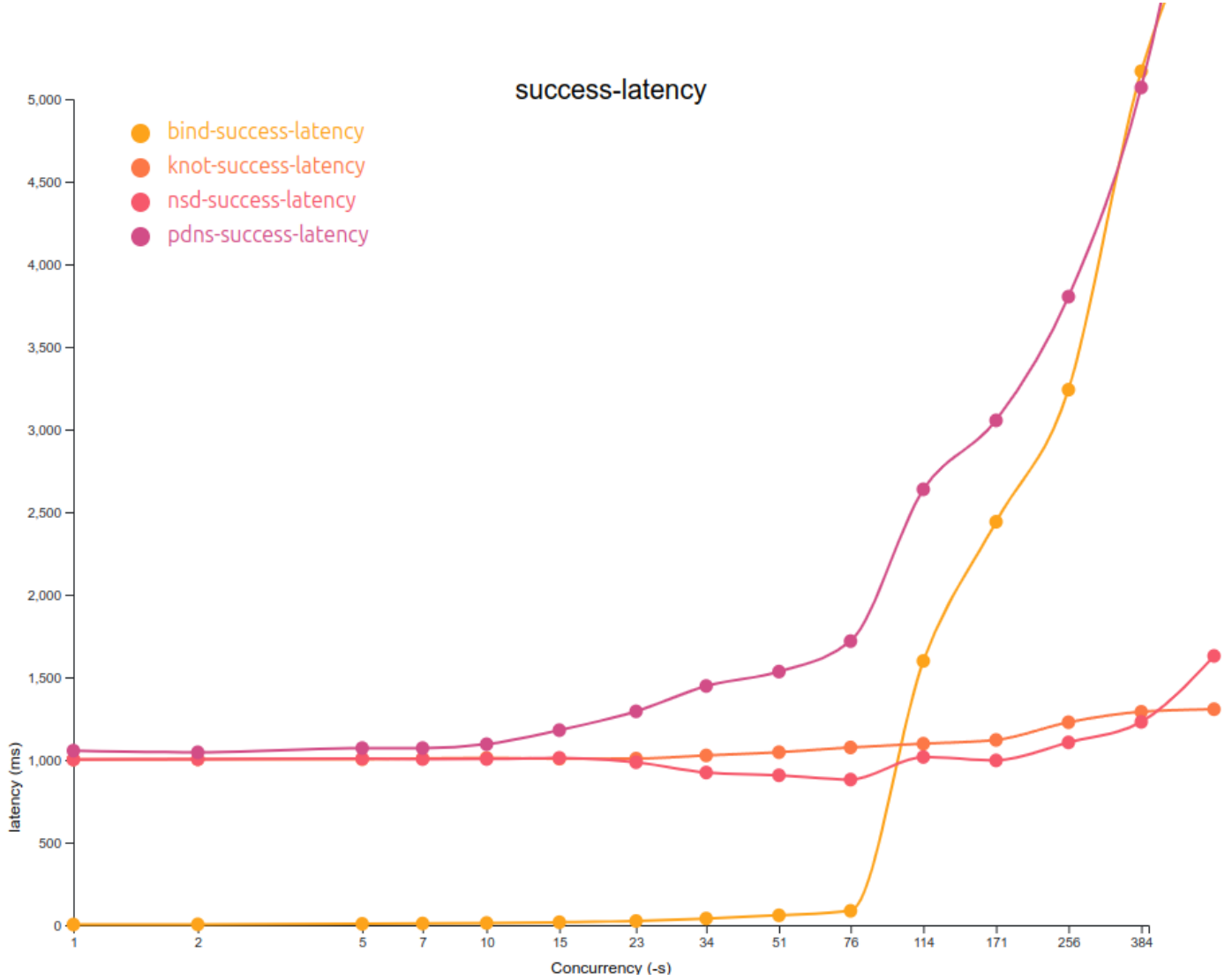


ULTRADNS®

Average time to transfer

- how long does it take from notify time to test xfr completion

- echo server ms timestamp trick
- kind of nice up to 76

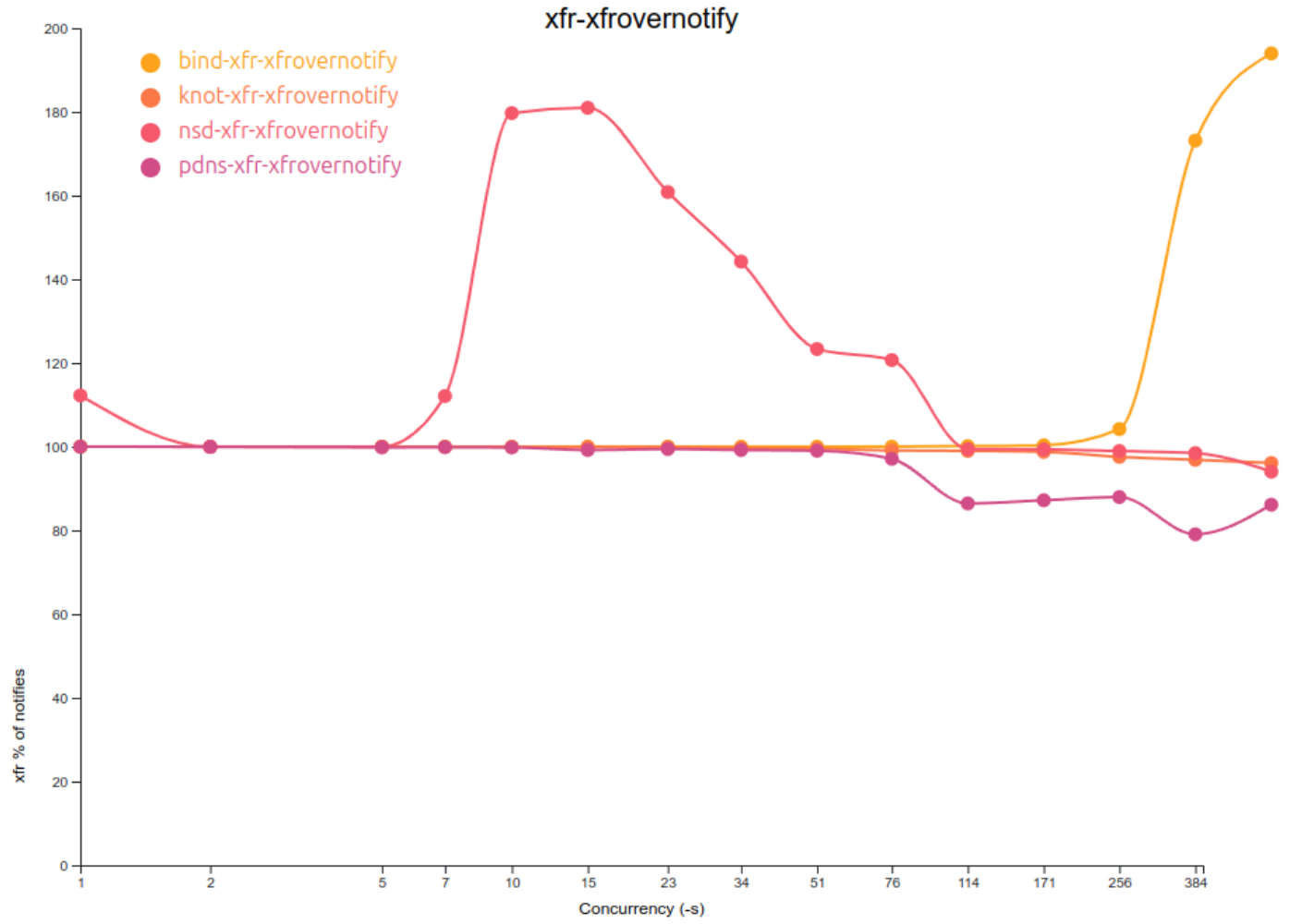


ULTRADNS®

XFR transactions per notify

- how many xfrs do we get per notify?

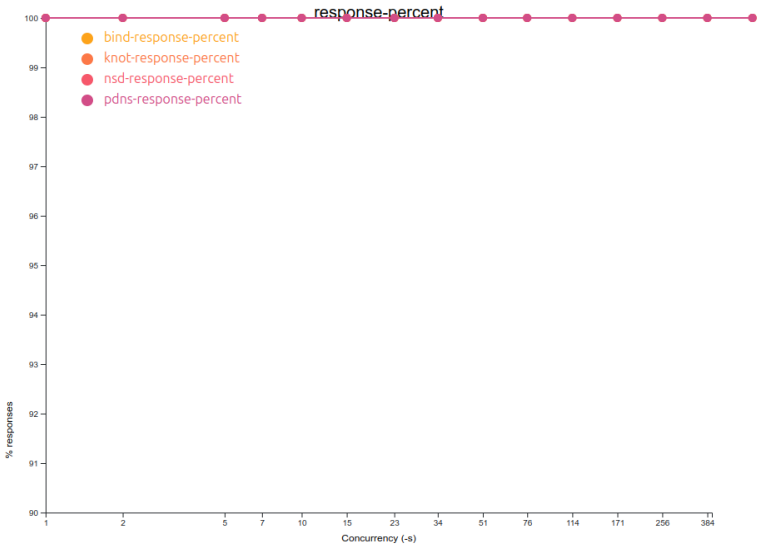
- nsd is a bit wild in this data
- refresh timers are difficult
- effects of concurrency are sensible (otherwise)

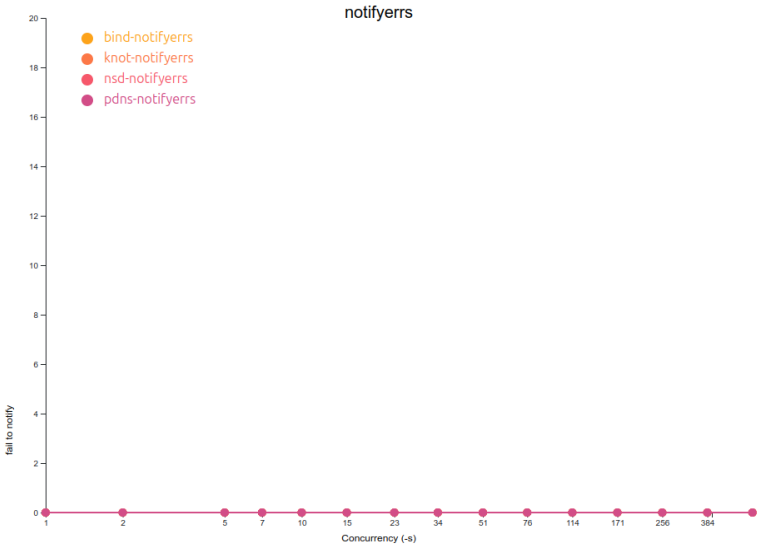


ULTRADNS®

Losing transfers

- validated the test environment
  - match up log lines to stats
  - 100% forwarded notifies
- receiving no notify errors
- receiving no xfr errors
- no nameserver error log lines!



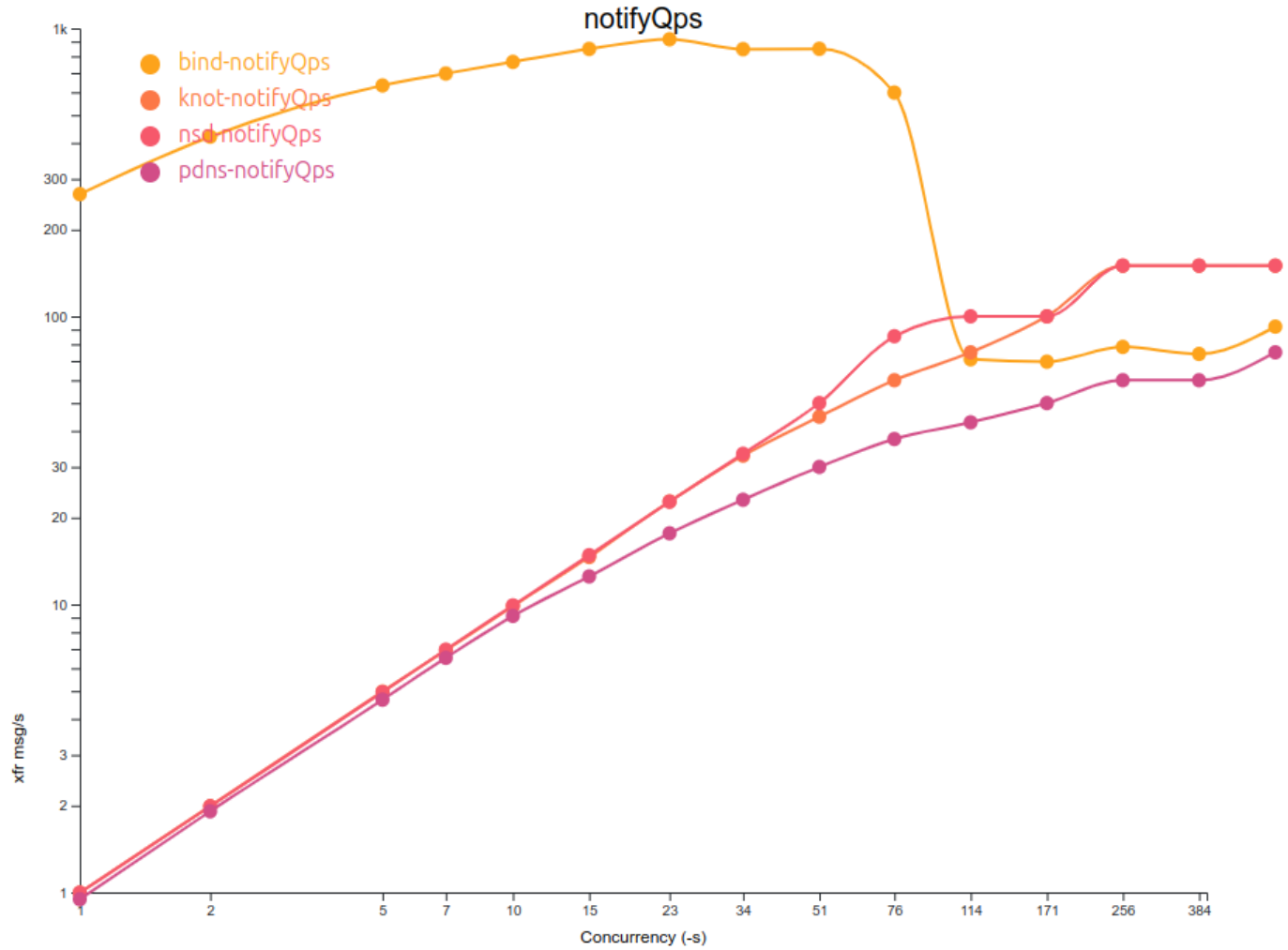


digicert

ULTRADNS

Rate limiting in action

- the nameservers other than bind will only do 1 transfer/second per zone



ULTRADNS®

Catalog Zone Updates

Add and remove zones

ULTRADNS®

What happens when we add and remove zones?

- this is a more complicated state machine – what to include?
- for simplicity – just the catalog for now

- Why are you doing this? Why???

What happens when we add and remove zones?

- tweak the echo server
- needs a few zones named after the timestamp
- send notify traffic

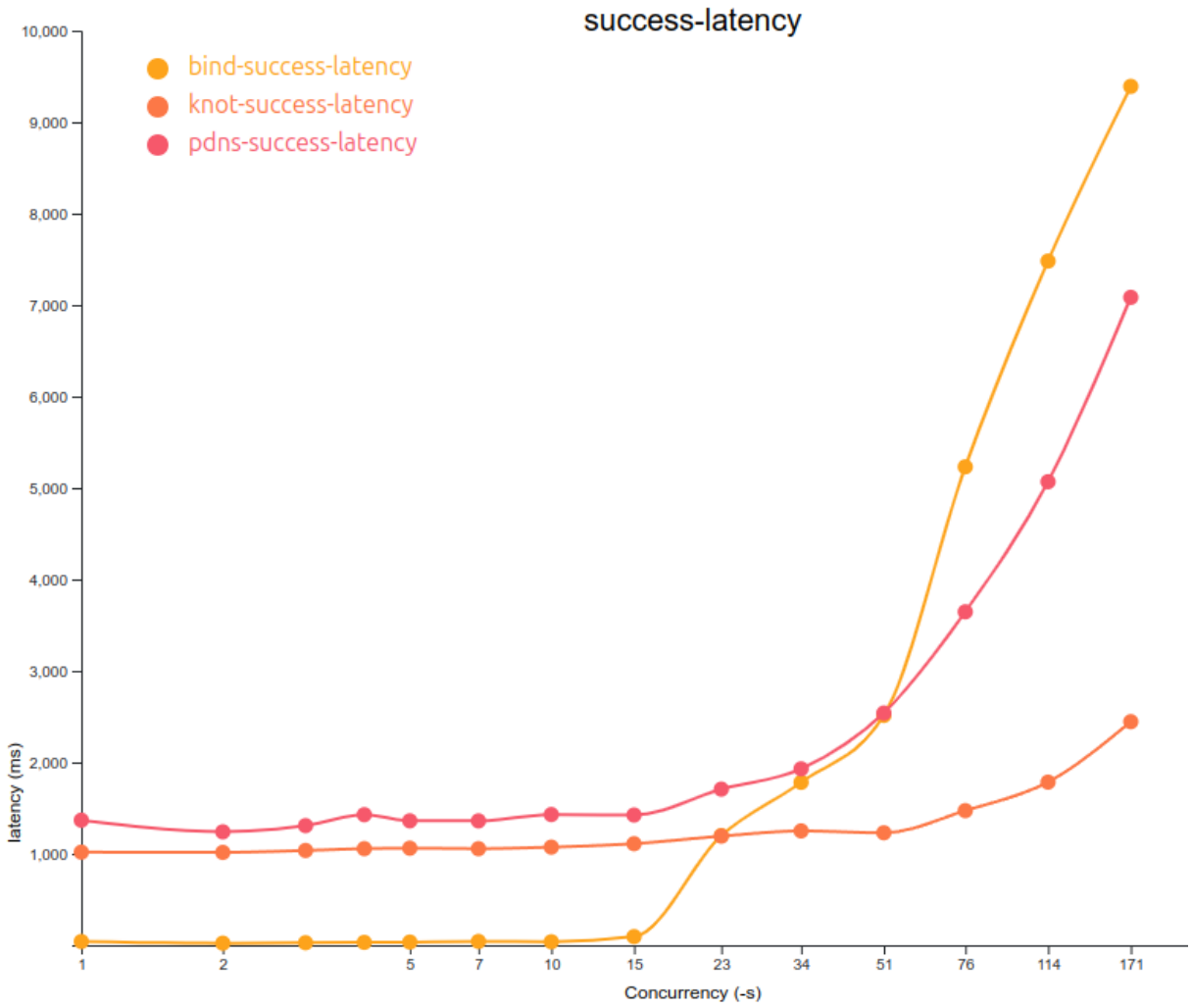
- The previous install isn't going to work, because there is only one catalog

- nsd doesn't do multiple catalogs

Catalog update performance

- time from notify to transfer out of the catalog zone (only)

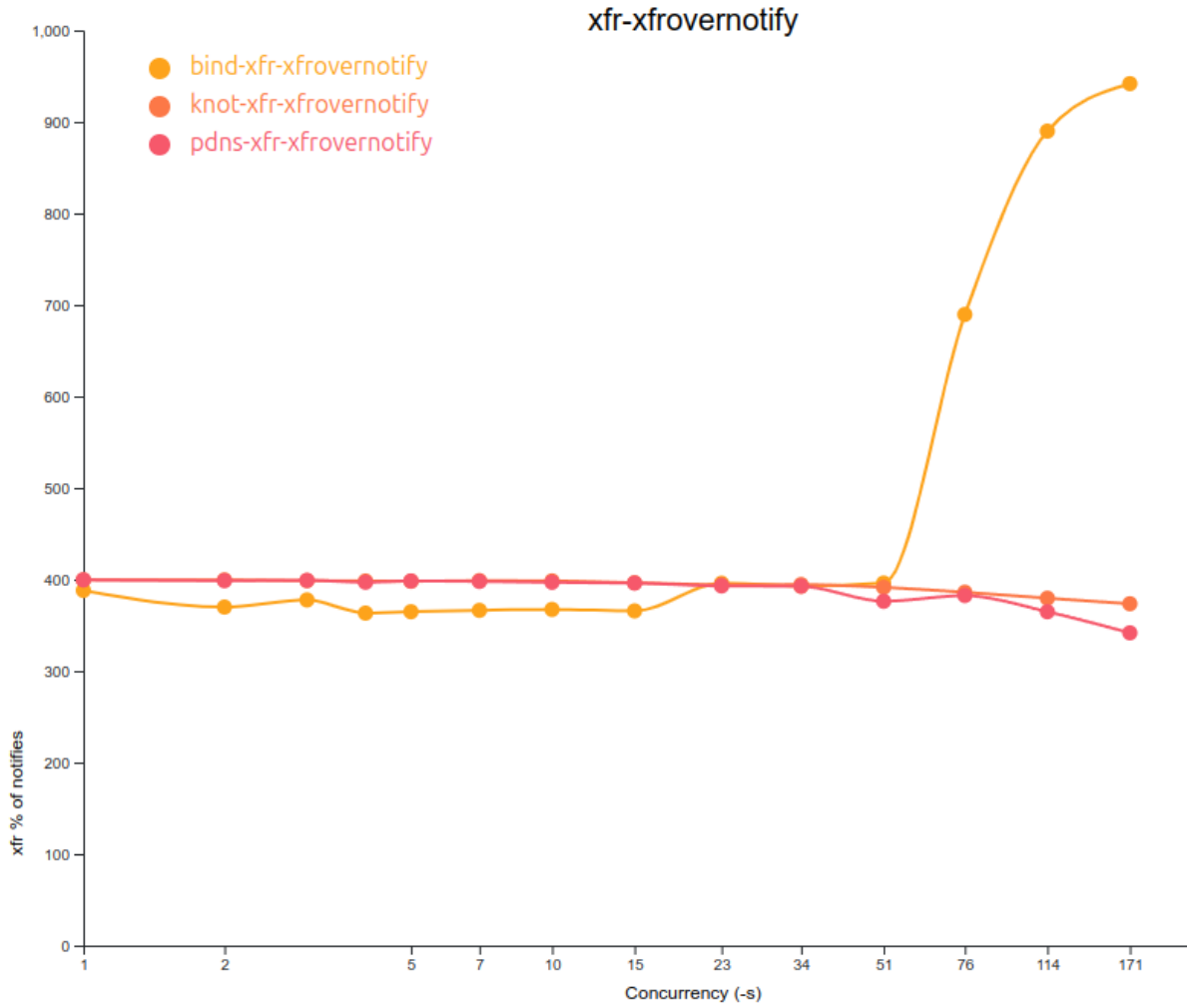
- not as performant as zone updates
- testable!



ULTRADNS®

Catalog update performance

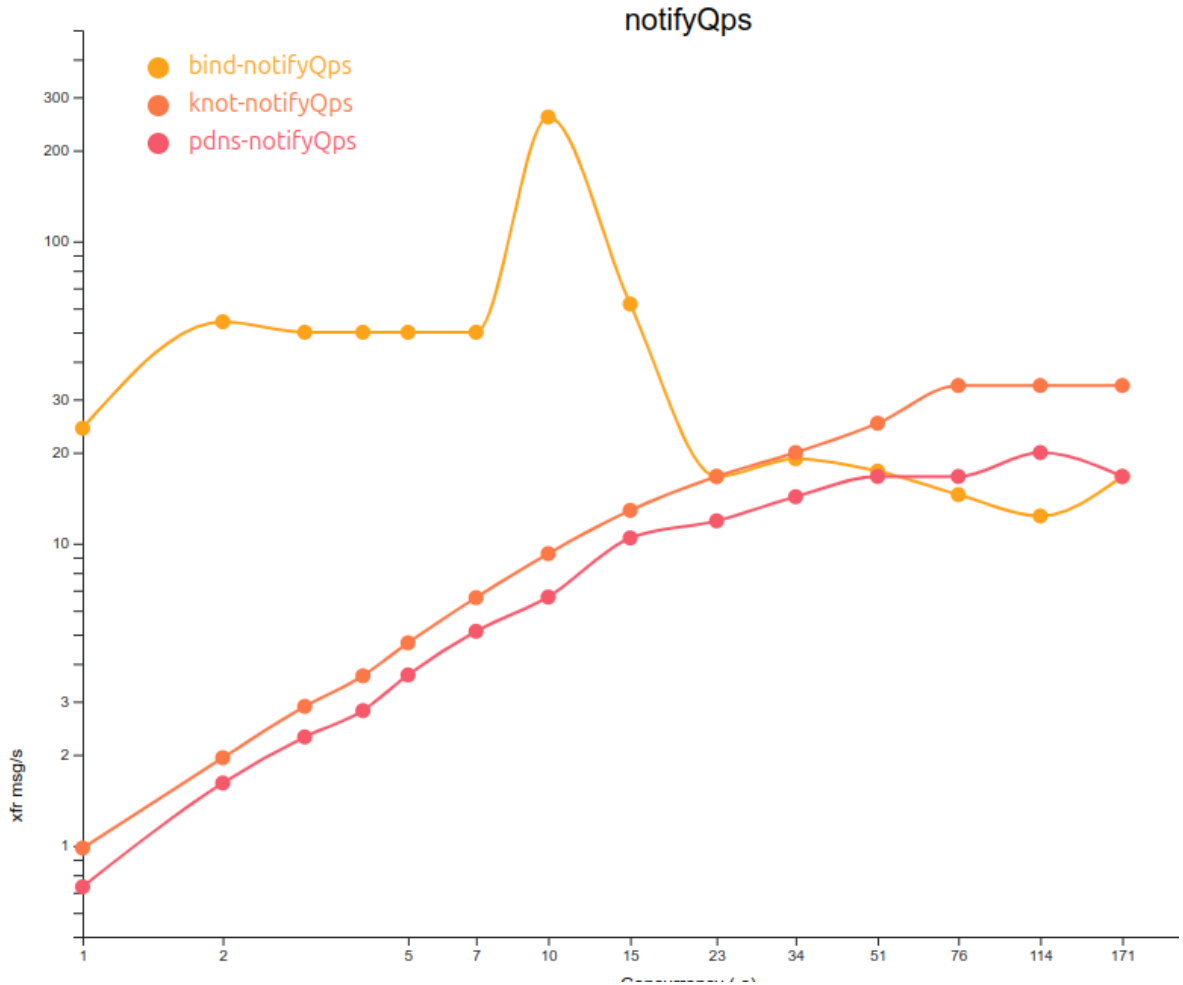
- ideally 4:1 notify:transfer
- each catalog has 3 members



ULTRADNS®

Catalog update performance

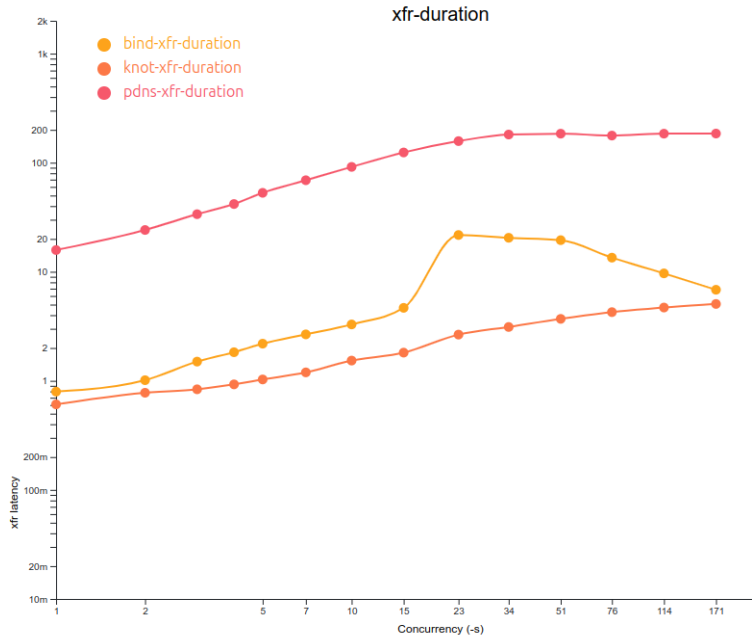
- average rate to complete 1k catalog updates
- what's up with bind?

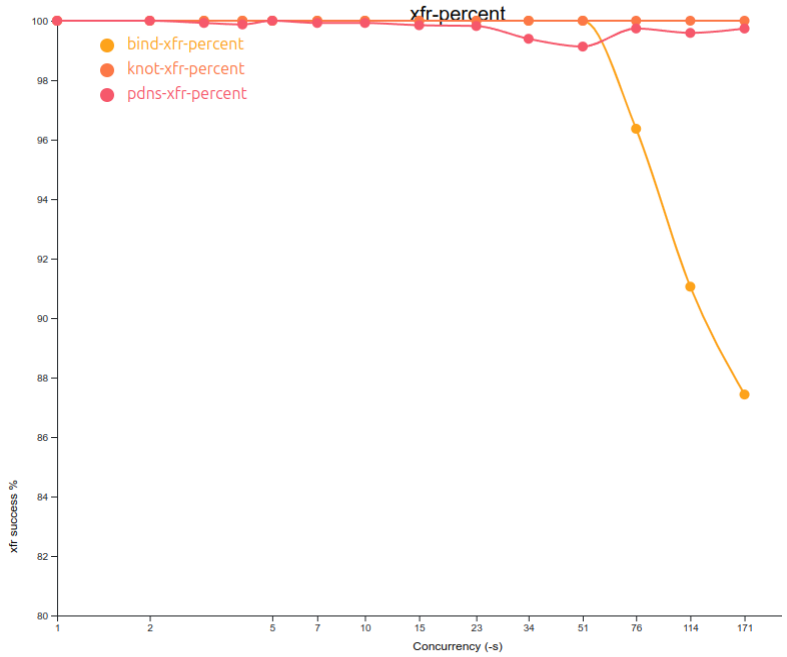


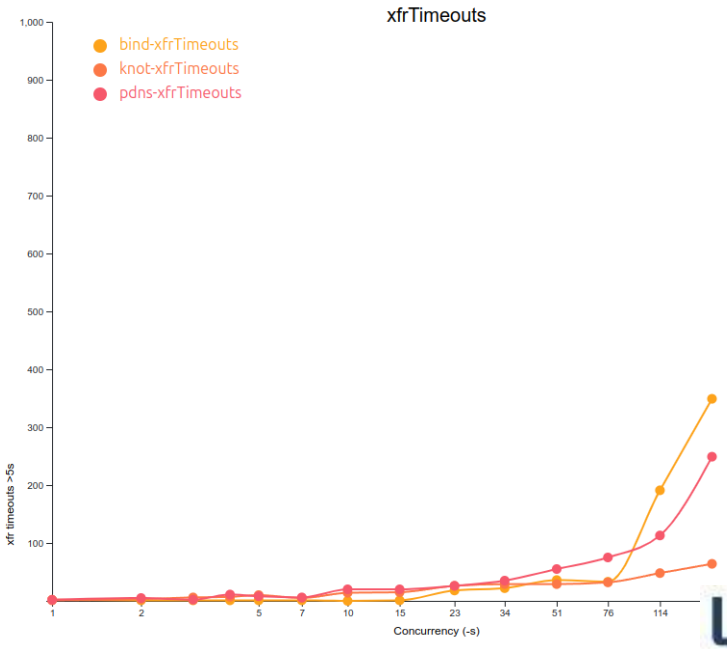
ULTRADNS®

Catalog update performance

- more than expected transaction timeouts
- xfr time was as expected







ULTRADNS®

Conclusions

Do we like zone transfers?

It depends

- this all seems a lot more complicated than db replication
  - on purpose!
- maybe that's what you need in your installation, or not
- installation-specific tuning is important

Future work

- need to deal with the refresh time by caching serial numbers
- multi-primary tests, failure scenarios
- simulated network latency
- more detail in the catalog updates, including member transfer time
- improved test performance

Thanks

This was fun

ULTRADNS®