



DNS Rex

Do you need an aggressive benchmark?

Alex Rousskov

The Measurement Factory

DNS Rex At a Glance

- A performance test tool for DNS resolvers.
- Born 2009 A.D. (Cenozoic Era).
- Designed to intimidate powerful resolvers.
- Could also quickly poison caching resolvers.
- Mission accomplished!
- Publicly released and dormant since 2012.
- Will fossilize without demand and more work.

Why a yet another DNS benchmark?

We said “no” more than once, but...

- Most tools focused on authoritative servers.
- Also needed to test cache poisoning defenses.
- Most tools were slow, unreliable, or shady.
- Angst and distrust among resolver engineers (see Exhibit A).
- Experience creating HTTP performance tools; it was “easy” for us to detect/foresee problems.

Exhibit A: Testing Resolver X

- Tool A's conclusion:

Maximum throughput: 22'346 qps

Lost at that point: 24%

- Tool B's conclusion:

Sustained throughput: 120'000 qps

Transaction errors: 0

- Argh!..

**Why a yet another DNS benchmark?
We said “no” more than once, but...**

... gave up and wrote what we needed.

Why no progress since ~2007? (a speculation)

- Easy problems have been solved (in 3K LOC):
 - send UDP queries at an increasing rate
 - bail on errors
 - RELEASE_NOTES:

January 10, 2008

Known Issues:

- None.

... Since ~2007

- No more automagic performance improvement!
 - MUST use threads for reasonable scale
- Remaining problems are much harder:
 - fundamental benchmarking problems
 - threading is difficult enough on its own
 - solving hard problems while threading is harder
- Past tool suppliers have to focus on survival.
- Insufficient demand???

... but if we want to move forward

What would an **ideal** tool for measuring caching resolver performance be?

Ideal: Persistence

sustaining load for longer than a few minutes

“The 3 million record query file has been replaced with a 10 million record query file as 3 million records were not enough for a full run on modern hardware.”

-- 2012 testing instructions

$$10\text{M} / 100\text{K QPS} = 100 \text{ seconds}$$

Ideal: Persistence

sustaining load for longer than a few minutes

“The longest single attack lasted nine days and 11 hours.”

-- NSFOCUS DDoS Threat Report

Ideal: Scalability

- SMP Scalability:

“faster” than any resolver on similar hardware

but since there is custom and \$\$\$ hardware...

- Swarm-ability:

test synchronization and results aggregation

across off-the-shelf and/or cheaper drones

Ideal: Scalability

worst case scenario?

“The single largest attack [rate was] 23 million PPS.”

-- NSFOCUS DDoS Threat Report

Ideal: Cache Awareness

ability to offer any configured hit ratio

- offered hit ratio is a ratio of hits that would be served by a perfect infinite cache
- relatively short traces: 100% offered hit ratio
- infinitely long traces: $X\%$ offered hit ratio

Ideal: Slowness

simulating authoritative server problems:

- response delays
- packet drops
- NXDOMAIN
- bad referrals
- errors

Ideal: Independence

- no 3rd party authoritative servers:
 - slow (what are you testing?)
 - difficult to configure correctly for the test
 - difficult to replicate
 - limited statistics
 - the real ones do not want to be attacked
- no resolver libraries?
- no resolver developers???

Ideal: Protocol Features

- IPv6
- TCP
- DNSSEC (1000s of generated signed zones!)
- NXDOMAIN (hijacking infrastructure tests???)

Ideal: Ease of use

- configuration files
- awareness and assessment of test environment
- detailed performance reports
- GUI???

Ideal: Other

Not detailing several key properties/features:

- reliability (but see Exhibit A)
- realism
- cost
- flexibility (scriptability??)
- portability
- openness?

DNS Rex vs The Ideal

(marketing)

- ✓ Reliability
- ✓ Persistence
- ✓ Scalability
- ✓ Cache Awareness
- ✓ Slowness
- ✓ Independence
- ✗ IPv6
- ✗ TCP
- ✓ DNSSEC
- ✓ Ease of use

DNS Rex vs The Ideal

(reality)

- $\frac{3}{4}$ Reliability Rex needs more exposure/testing to be sure
- ✓ Persistence
- $\frac{1}{2}$ Scalability Rex supports SMP scale but not swarming
- $\frac{3}{4}$ Cache Awareness only 0% and 100% hit ratio is configurable
- $\frac{3}{4}$ Slowness configurable think time but not error ratio
- ✓ Independence
- ✗ IPv6 mostly ready but lacking configuration
- ✗ TCP
- $\frac{1}{5}$ DNSSEC sends DO but relies on manual zone signing
- $\frac{1}{2}$ Ease of use Rex has config file, detects overload, but ...

What's Next?

- Leave DNS Rex as is, allowing it to die?

or

- Relaunch the project?
 - focusing on what features?

Feedback

Alex Rousskov

The Measurement Factory

info@measurement-factory.com

<http://rex.measurement-factory.com/>