



Knot DNS



High-performance authoritative DNS server

Ondřej Surý • ondrej.sury@nic.cz • 23.05.2013

What is Knot DNS?

- Authoritative(-only) DNS server
- Open source – GPL v3
- Current stable release: v1.2.0



Project history

- Fall 2011 – Public beta release
- Spring 2012 – Knot DNS 1.0
- Fall 2012 – Knot DNS 1.1
 - Massive transfer (IXFR) speedups
 - Reference manual!
- Spring 2013 – Knot DNS 1.2
 - Dynamic updates, Response Rate Limiting



Design goals

- Open-source authoritative-only DNS server
 - Developed in an open way (we do not hide our mistakes)
- Usable for everybody (root zone, TLDs, massive DNS hosting)
- Fast(est), feature-rich
 - Very close to the theoretical limits of the platform (trafgen)
 - 250k-300k on commodity hardware
- On-the-fly reconfiguration
- Portable, modular
 - Linux, *BSD, MacOSX
 - Portability mainly depends on userspace-rcu library



Standards compliant

- AXFR/IXFR (both master and slave)
- All known RR Type support
 - Including TYPE#nnnn
- DNSSEC with NSEC3
- TSIG supported
- Root zone support
- NSID support (RFC5001)
- Fast track new standards
 - DANE Protocol (TLSA RR)



Configuration

- Curly braces and semicolons (C-like)
 - Interfaces (IPv4 or IPv6)
 - Remotes (masters or slaves)
 - Zones
 - Keys (TSIG)
- Logging (syslog or file-based)
- Precompiled zones (removed from 1.3.0)
 - Offload the parsing from main server
 - No longer needed with new parser



Knot DNS design

- Minimize amount of lookups for one query
 - Optimized zone structures
 - References to related data
- Minimize lookup time
 - Hash table with worst-case $O(1)$ lookup time (Cuckoo hashing scheme)
 - Lock-free architecture
- Non-stop operation, run-time updates
 - Read-Copy-Update (always consistent data)
- Copy-on-Write (shallow copies)



Constantly aiming for excellence

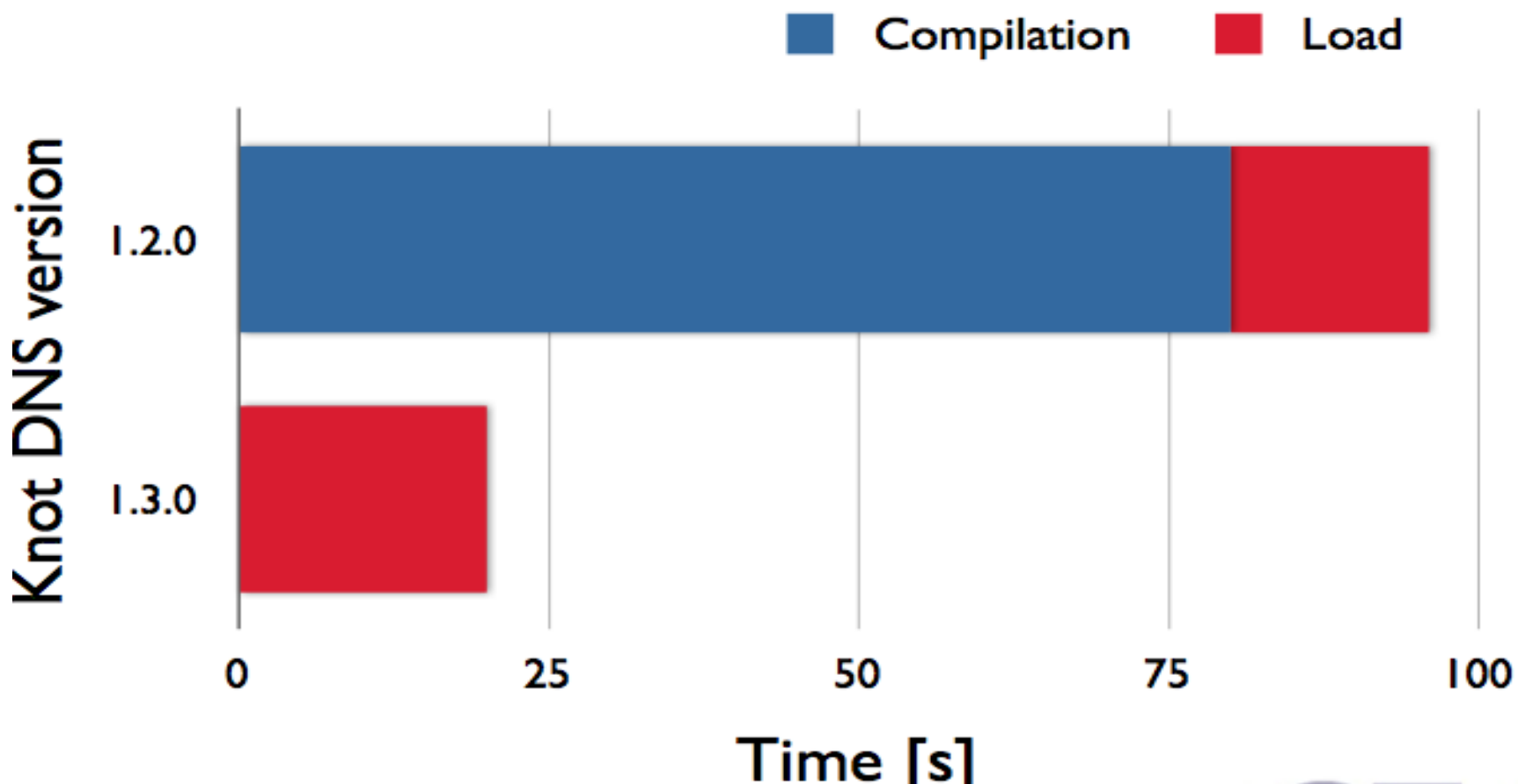
- Knot DNS 1.3.0 (Internal pre-release)
 - Improved networking code
 - New parser written in Ragel
 - Lower memory requirements
 - Better allocation
 - Data structures optimizations
 - New utilities: kdig, khost, knsupdate



Knot DNS 1.3.0 – new parser

Box - Xeon E5-2609 @ 2.4 GHz (Linux 3.2.0)

Zone file - 563MiB, ±5 000 000 RRs



Memory consumption in 1.3.0

- 600MB zone with DNSSEC
- Knot DNS 1.2.0
 - 32bit: 1115MB
 - 64bit: 2052MB
- Knot DNS 1.3.0
 - 32bit: 1022MB (100MB less)
 - 64bit: 1500MB (550MB less)



Future plans

- DNS Control Protocol
 - Interoperable with other DNS servers (Bind, NSD, ...)
 - Client-server
 - Master-slave
 - DNSSEC signing
 - One-click solution
 - Further optimizations
 - Speed
 - Memory
 - Knot DNS Resolver
- We want to hear you



How do we test?

- Unit tests
- Static source code scan
 - Coverity Scan program (<http://scan.coverity.com/>)
 - LLVM Clang
- Query replay testing
- BASH/Python testing – malformed packets, ...

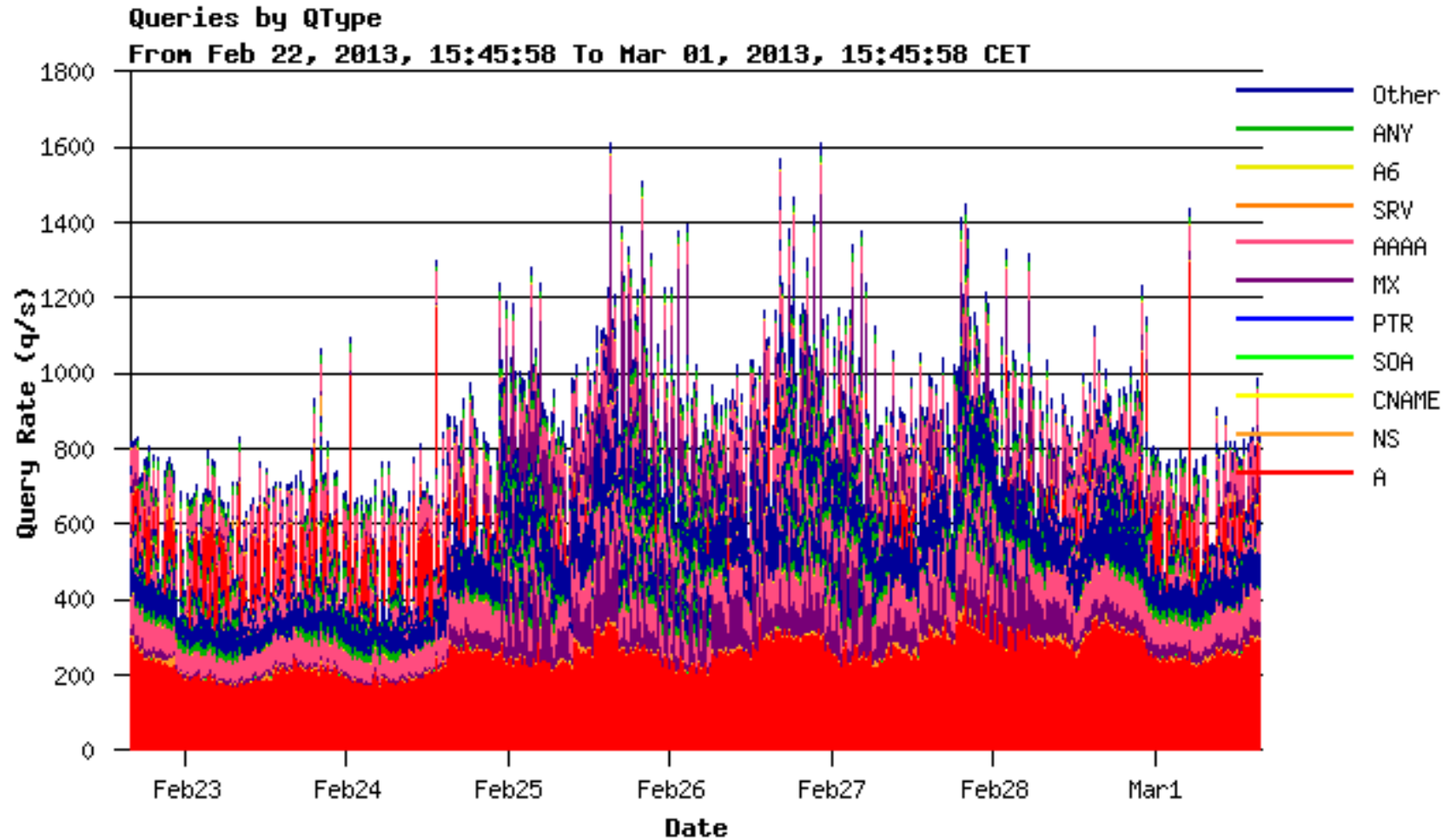


Known deployments

- .CZ (+ several small zones) – anycast cloud
- .DK – one slave
- hosting90.cz, igloonet.cz (50-100k domains)



.cz – Knot DNS node traffic



Planned deployments

- L-ROOT (ICANN)
 - Huge anycast farm for l.root-servers.net.
- .RU
- Other gTLDs: .UA, .SE, .UK
- Registrars



Pre-packaged Knot DNS

- Linux
 - Debian
 - <http://packages.debian.org/knot> (jessie,sid)
 - deb <http://deb.knot-dns.cz/debian/> {squeeze,wheezy} main
- Ubuntu
 - <http://packages.ubuntu.com/knot> (quantal, raring)
 - [ppa:cz.nic-labs/knot-dns](http://ppa.cz.nic-labs/knot-dns) (all supported releases)
- Fedora
 - Knot DNS is part of Fedora Project
 - <https://admin.fedoraproject.org/pkgdb/acls/name/knot>
 - <http://rpm.knot-dns.cz/redhat/>
- OpenSUSE
 - OpenSUSE Build Service / OpenSUSE Factory
 - <http://software.opensuse.org/package/knot>
- FreeBSD
 - <http://www.freebsd.org/cgi/cvsweb.cgi/ports/dns/knot/>

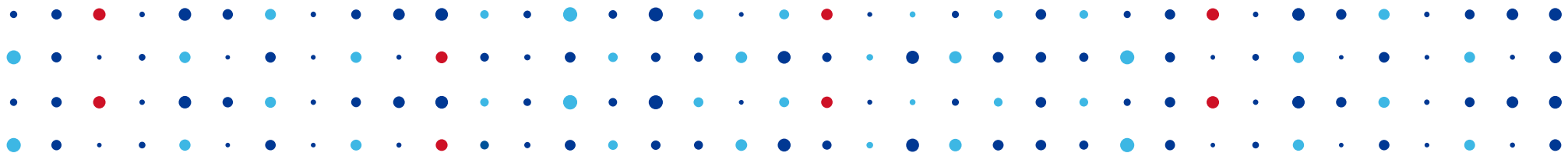


Resources

- Home page: <http://www.knot-dns.cz/>
- Google+ page with news: <http://goo.gl/f7IWF>
- Issue tracking and source code
 - Contributions welcome!
 - <https://redmine.labs.nic.cz/>
 - <git://git.nic.cz/knot-dns>
- Mailing list

knot-dns-users@lists.nic.cz





Thank You

Ondřej Surý • ondrej.sury@nic.cz

