# BIND 9.12 Refactoring and Performance Advances

September 2017

Evan Hunt
each@isc.org

# Where we came from

- BIND 9 project started 1998
- Now approaching half a million lines of code
  - 3x PowerDNS
  - 5x Unbound
  - 6x Knot authoritative
- Decisions were made that need revisiting:
  - Hardware, memory, and DNS assumptions from circa Y2K
  - Function design that didn’t extend cleanly as new features were added
  - Module design doesn’t afford testability

# Example: query_find()

- Implements query processing logic
- 800 LOC and multiple goto statements in original BIND release (i.e., already a hairball)
- 2400 LOC and *more* gotos in BIND 9.11 (2016).
- McCabe Complexity: **468** (20-30 is considered high)
- Original query logic, plus:
  - dns64
  - RPZ
  - RRL
  - NXDOMAIN redirection
  - Prefetch
  - Etc...

# Example: resquery_response()

- Handles responses from authoritative servers
- 400 LOC in original BIND release
- 1100 LOC in BIND 9.11 (2016).
- McCabe complexity **175**
- Original logic, plus:
  - EDNS error handling
  - Other exceptional cases
  - Statistics
  - DNSTAP
  - Etc...

# Testability Issues

- BIND has extensive system/integration level testing
  - over 100 system tests with many hundreds of subsidiary test cases)
  - ~45,000 lines of test code in shell/perl
- Ongoing fuzz testing (thank you AFL)
- Ongoing performance testing (thank you Ray)
- We have been adding unit tests in newly added library code since ~2010.  BUT:
  - Much of the query processing is implemented in the named binary, not in libraries...
  - So in addition to functions being too big to reasonably unit test, many are not in a place that a unit test can link to

# Where we are

- In 2016, Witold Krecicki and I began a project to break up the largest functions and reduce their complexity
- I started a project to move query functions from named to libns so unit tests (when written) can link to them

# resquery_response()

- Broken into ~30 smaller functions
  - Most have less than 100 lines of code
  - Most are under 20 McCabe complexity
  - Worst remaining complexity 68
- Added comments detailing call flow
- Unit tests to come

# query_find()

- Moved into libns
- Broken into ~35 smaller functions
  - Most have less than 100 lines of code
  - Most are under 20 McCabe complexity
  - Worst remaining complexity ~50
- Added comments documenting call flow
- Unit tests have been started (but still minimal)

# Testability

- Still an ongoing process, but we’re in a better position for unit tests of name server code
- System test code coverage in affected functions is above 80% (mostly lacking pathological cases)

# Performance

- BIND systems for memory management, task management, etc, were designed with assumptions no longer valid
- Years of new features have been added without measurement of performance regression (thanks to Ray Bellis for addressing this with perflab!)
- In 2016, Mukund Sivaraman began a project to identify bottlenecks and address them.

# Performance (cont'd)

- BIND was particularly weak (compared to other name servers) with delegation-heavy zones such as root and TLDs
  - Lots of rdata lookups per response
  - minimal-responses helped a lot
  - acache helped a little
- Even in non-delegation-heavy operation, there was inefficiency

# Performance work done

- Refactored several basic functions:
  - Name compression
  - Name capitalization
  - Hashing
  - Buffer operations
- Turn on minimal-responses by default
- Removed acache; replaced with much more efficient glue cache (also on by default)
- Improve lock contention
- No specific RRset ordering
- Option to use system malloc
- Don’t fill memory by default

# Performance results

- When serving the root zone:
  - 9.11 (default settings): 63 kqps
  - 9.11 (acache, minimal-responses): 102 kqps
  - 9.12: 390 kqps
  - Speedup: factor 4-6
- When serving typical authoritative domains:
  - 9.11 (default settings): 540 kqps
  - 9.12: 674 kqps
  - Speedup: factor 1.25

# Questions