ISC KEA Performance

Wlodek Wencel, QA engineer
Eddy Winstead, Sr. Sales Engineer

April 22, 2020

https://www.isc.org
Webinar will be recorded

Recording will be posted ~ couple days, on ISC’s Youtube channel and on https://www.isc.org/presentations/

Questions - enter in Q&A or Chat panel, we will address these at the end
Webinar agenda

● When might you need higher performance?
● Previous performance tests
● Test set-up, methodology
● Results
● Plans for future testing
● Q&A
How much performance do you need?

Kea is not performance-constrained in most scenarios.

For example:

86,400 clients require only 2 LPS

assumptions: 24 hour lease duration, 12 hour renewal = 86400 DORA in 24 hours

of course, the renewals won’t be evenly spaced
Who might need more performance?

• Unusually high number of clients requesting at once - e.g. rebooting scenario
• Unusually short lease - e.g. public wifi
• Database backends slow performance in general, so more performance there is good
Implementing Multithreading

Developing in 1.7 development branch

- Partially complete in 1.7.6
- Still working on mt support for hooks
- Releasing as 1.8 stable version when complete
- Aiming for 1.8 release in summer 2020
- This test measures impact of multithreading vs single threading
Kea Performance Baseline

Published performance results from Kea 1.4 - 1.5
https://kb.isc.org/docs/kea-performance-tests-140-vs-150

Measured the gap between memfile and db backends.

NB: KB on performance considerations
https://kb.isc.org/docs/kea-performance-optimization
Testing Set up

**system under test**

Dell R340
Kea 1.7.6, MySQL, PostgreSQL

**second system for HA testing**

Dell R340
Kea 1.7.6, MySQL, PostgreSQL

1 gig ethernet

MySQL databases had additional configuration:
inno$db_flush_log_at_trx_commit=2
Test Platform details

- Kea servers are running on 2 Dell R340 servers:
  - CPU Intel Xeon E-2146G 3.5GHz 6 cores/12 threads
  - 64GB RAM
  - 3 x SSDs 446GB each in HW RAID-0 configuration (virtual disk size 1338GB)
  - Intel(R) 10GbE 2P X710 Adapter (2 ports)
  - OS - Ubuntu 18.04.4 LTS
Traffic generator

• perfdhcp - open source
• Included in Kea sources
  • enabled with configure time switch
    (--enable-perfdhcp)
• And in packages (all on Cloudsmith):
  • rpm - isc-kea
  • deb - isc-kea-admin
• Documentation included.
Kea configuration

• One subnet with one large pool
• No host reservations, no client classification, no options
• Functions are enabled by default are still enabled for tests - we use default values
• Lease lifetime is longer than test duration
• Every time starts ‘clean’
• Threads number differ in tests
• Databases are always local
Traffic details

• Just DORA/SARR (no renew, rebinds, releases)
• Each client requests an address once though entire test
• Client request does not include any extra options, only those essential to get an address
• This is not intended to be a ‘realistic’ example of a production scenario.
Test Design

• Measure performance with 2.5% packet ‘drop rate’
• Packet is considered dropped when response is > 2 seconds
• We have two types of results:
  • lease rates (results are the average of middle 7 results of 9 test runs)
  • observing second by second how Kea works
Why 2.5% drop rate?

• calculated rate is used later for other, long running tests and:
  • if we use 0% drop rate it would not reveal cumulative /systematic problems
  • flooding is not helpful
• We tested at 1%, 1.5%, and didn’t see enough stability in the results
• With 2.5% drop rate, results were consistent and we thought this is an acceptable drop rate in production.
Basic results

1. Kea v4 backends single thread.
   - memfile: 11248
   - mysql: 1990
   - postgresql: 1778

2. Kea v6 backends single thread.
   - memfile: 13028
   - mysql: 2103
   - postgresql: 2135

3. High Availability v6 single thread.
   - hotstandby: 1096
   - loadbalancing: 3614

   - memfile: 19695
   - mysql: 5712
   - postgresql: 6562

5. Kea v6 backends multi threading.
   - memfile: 36811
   - mysql: 4562
   - postgresql: 7062

6. High Availability multi threading
   NOT SUPPORTED YET
Observations

• Performance improved 190 - 350%
• Used thread number = 4 for memfile
• Used thread number = 6 for PostgreSQL
• Used thread number = 12 for MySQL
• DHCPv6 is much faster - to be investigated
• HA with multithreading support in 1.7.8
• Hyperthreading may affect performance
Kea + pgsql, single and multithreading

35. test pgsql v6 second by second

37. test pgsql v6 multi threading second by second
Observations

• As Kea is faster, it exposes the performance limits of the db backends - with multithreading, we have multiple db connections
• Pauses in response are due to database writes, and it’s duration and affect to kea differ
• Database tuning should be useful here - we did not attempt that in this test.
What happens when pool utilization is too high?

Pool utilization reached 96% here
Interactive report

• html file with all of these results is attached to the kb article

Also includes
• results history (multiple runs with multiple versions)
• stability when Kea combined with db backend.
• more details
  • min, max, average delay
This test is still running....

- Current results can be found at our Jenkins page: https://jenkins.isc.org/job/kea-1.7/job/performance/KeaPerformanceReport/
- New test runs are triggered with code commits, and since this is testing the master branch, performance may change slightly from one run to the next.
Questions for future testing

● How does host reservation affect performance?
● How does client classification affect performance?
● How do huge number of subnets/pools affect performance?
● How does hyperthreading affect Kea performance?
● Experimenting with database tuning
● ‘Avalanche’ scenarios
● More realistic test scenarios
Tips and tricks for testing Kea by yourself

- Our configuration files included in KB article
- perfdhcp commands with corresponding config
Thank you!

• Full results at: https://www.isc.org/kea-performance/
• KB article on this test: https://kb.isc.org/docs/kea-performance-tests-17-multithreading
• Software downloads: https://www.isc.org/download or https://downloads.isc.org
• Packages at: https://cloudsmith.io/~isc/repos/kea-1-7/packages/
• Presentations: https://www.isc.org/presentations
• Main GitLab: https://gitlab.isc.org