Which habitat fits your name server's nature best?

Findings while measuring NSD4

Willem Toorop

Willem@NLnetLabs.nl

NLnet Labs

16 October 2013

What is this about

- Performance measurements for the upcoming NSD4
- Comparison tests:
  - UDP/TCP queries per second (with 1 - 4 CPU cores/threads/processes)

    (On Linux 3.9 and FreeBSD 9.1)
  - Memory usage
- Name servers:

Bind	9.9.2-P1	
NSD	3.2.15	
NSD	4.0.0b4	
NSD	4.0.0b5	
Knot	1.2.0	
Yadifa	1.0.2-2337	
PowerDNS	3.3	(TCP qps only)

What is this about

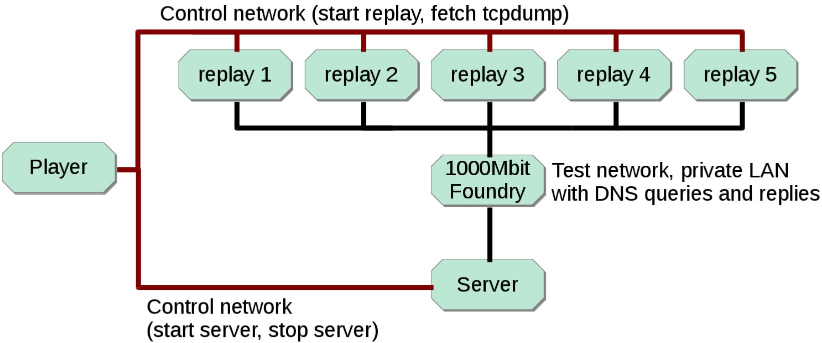
- Performance measurements for the upcoming NSD4
- Comparison tests:
  - UDP/TCP queries per second (with 1 - 4 CPU cores/threads/processes)
    (On Linux 3.9 and FreeBSD 9.1)
  - Memory usage
- Name servers:

Bind	9.9.2-P1	
NSD	3.2.15	
NSD	4.0.0b4	
NSD	4.0.0b5	
Knot	1.2.0	
Yadifa	1.0.2-2337	
PowerDNS	3.3	(TCP qps only)

- We noticed that different circumstances (number of CPUs, Linux/FreeBSD, Memory) suited different name servers differently

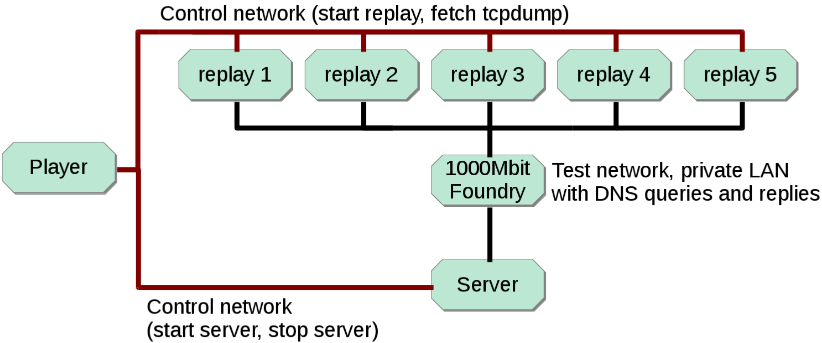
Performance measurements - Method and setup

- Domain Name Server Testing Lab (DISTEL)
- Foundry FastIron WorkGroup X448 1000Base-T
- Dell PowerEdge 1950, 2 x 64-bit dual-core Xeon 5130 2.00GHz, 4MB Cahce, 1333 MHz FSB, 8GB Ram
- on-board Broadcom NetXtreme II BCM5708 1000Base-T


Control network (start replay, fetch tcpdump)
replay 1
replay 2
replay 3
replay 4
replay 5
Player
1000Mbit Foundry
Test network, private LAN with DNS queries and replies
Server
Control network (start server, stop server)

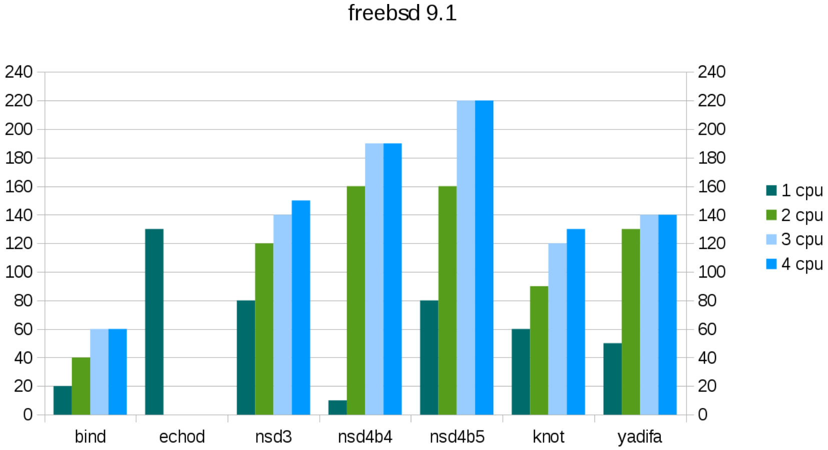
Performance measurements - Method and setup

- **D**omain Name **S**erver **T**esting **L**ab **(DISTEL)**
- Synthetic unsigned fake root zone with 500 delegations
- Queries in random order (no NXDOMAIN)
- Player directs the replayers for varying speeds


Control network (start replay, fetch tcpdump)
replay 1
replay 2
replay 3
replay 4
replay 5
Player
1000Mbit Foundry
Test network, private LAN with DNS queries and replies
Server
Control network (start server, stop server)

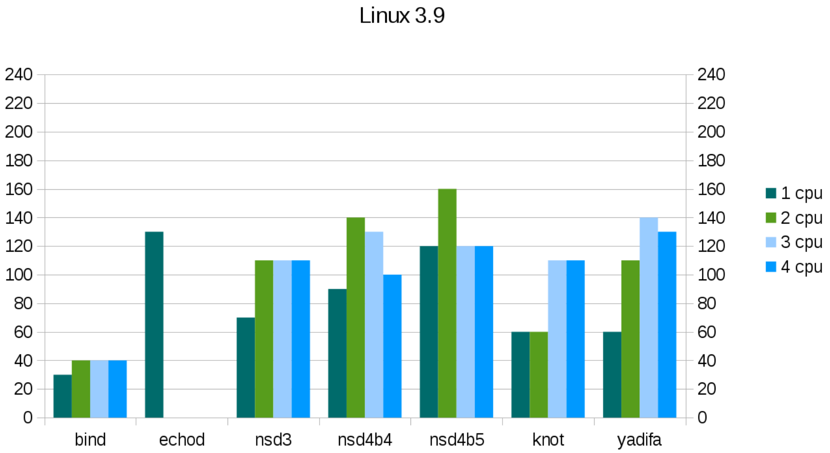
Performance measurements - Results

- thousands of queries per second



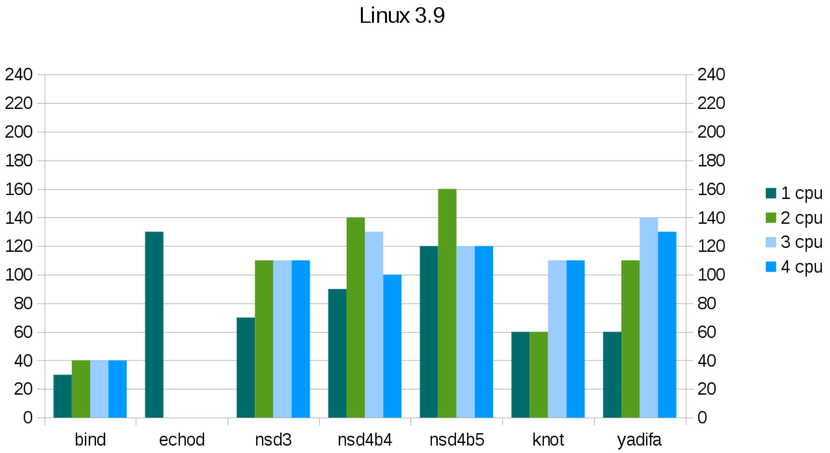
Performance measurements - Results

- ▶ thousands of queries per second



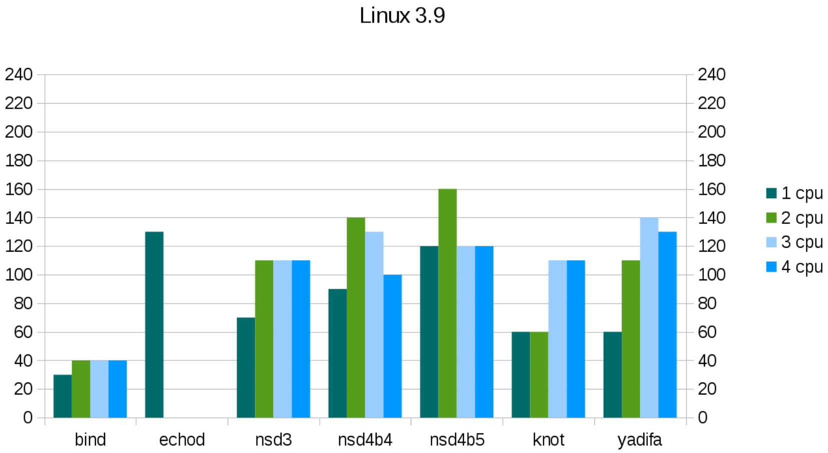
Performance measurements - Results

- Knot and Yadifa perform similar or better on linux when number of CPUs > 2



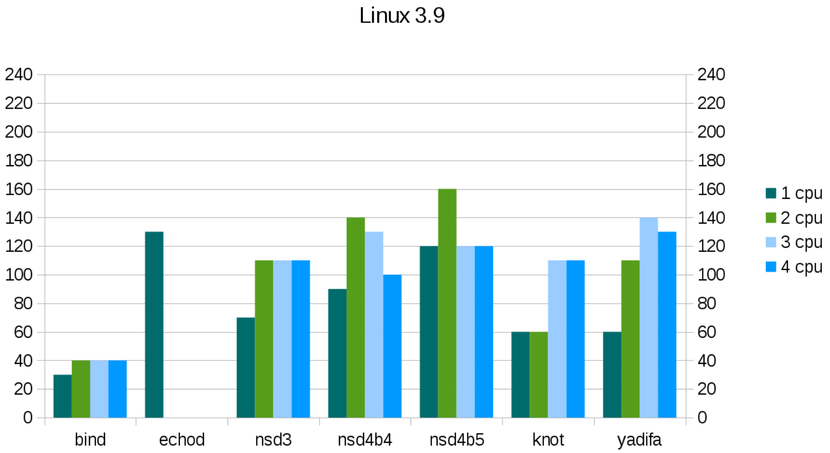
Performance measurements - Results

- Knot and Yadifa perform similar or better on linux when number of CPUs > 2
- Knot and Yadifa use threads, NSD is processes based
  Bind is compiled with threads support for comparison



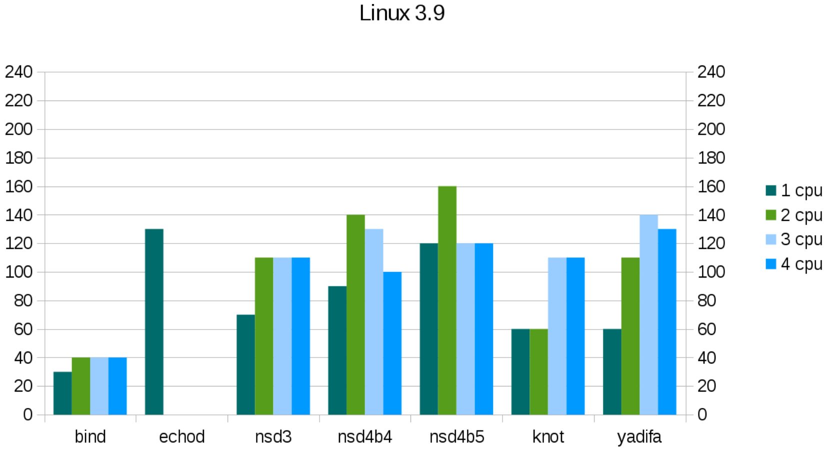
Performance measurements - Results

- Knot and Yadifa perform similar or better on linux when number of CPUs > 2
- **luka**: You have dual-cores. Perhaps a NUMA issue?



Performance measurements - Results

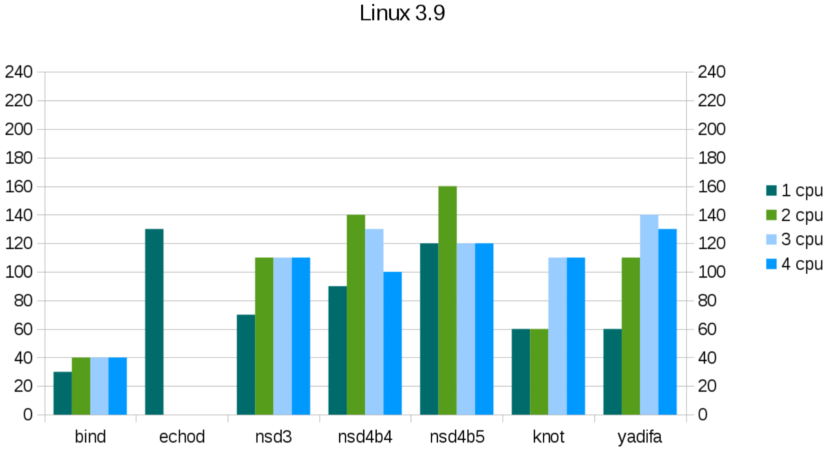
- Knot and Yadifa perform similar or better on linux when number of CPUs > 2
- In all cases the Linux interrupt handler `ksoftirqd` saturated all remaining cores



Performance measurements - Results

- With the test set up used, on Linux, for:

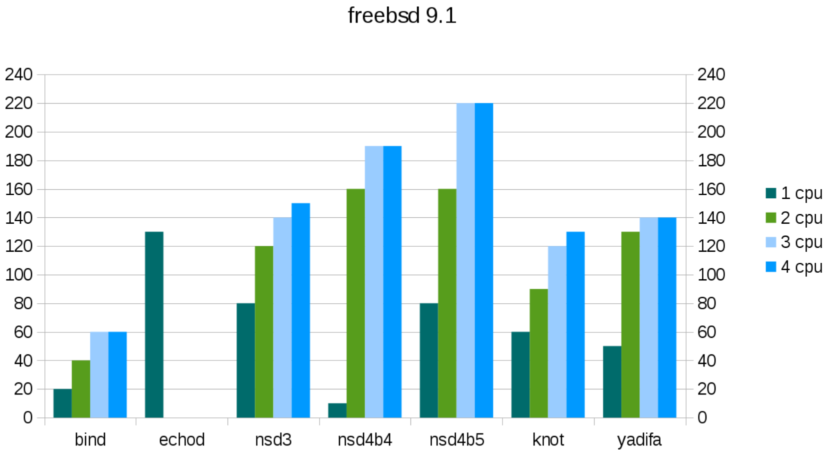
Bind & Knot	Use 4 out of 4 cores	(use all cores?)
Yadifa	Use 3 out of 4 cores	(use # cores - 1?)
NSD	Use 2 out of 4 cores	(use # cores - 2?)


Linux 3.9

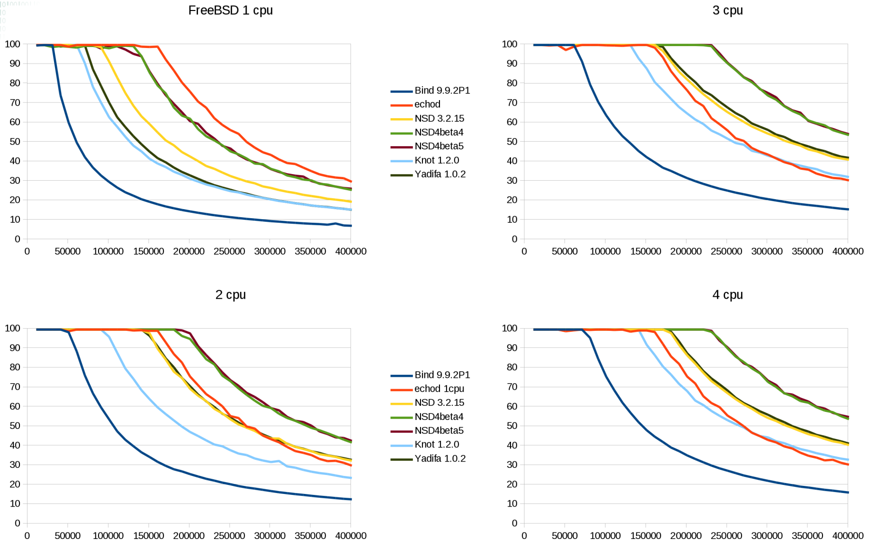
Performance measurements - Results

- With the test setup used, on FreeBSD, for:

Bind, Knot, Yadifa & NSD	Use 4 out of 4 cores (use all cores?)



Performance measurements - Results



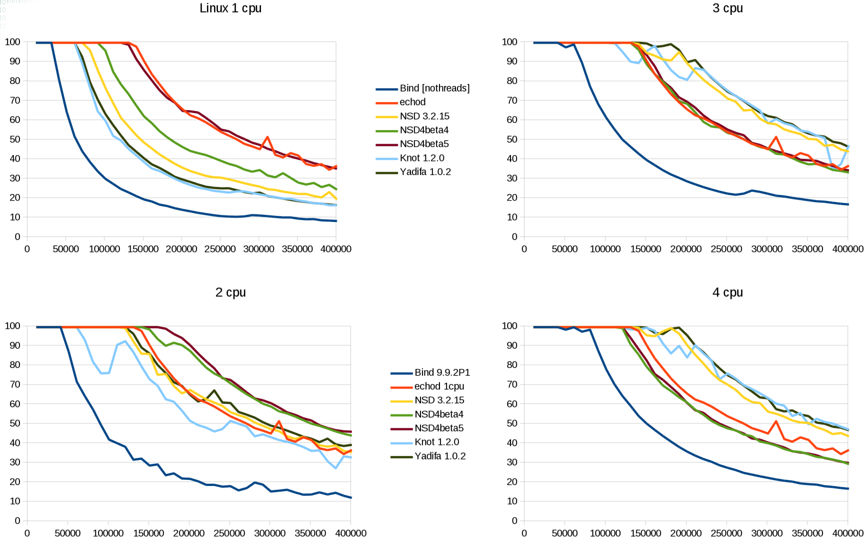
FreeBSD 1 cpu

3 cpu

2 cpu

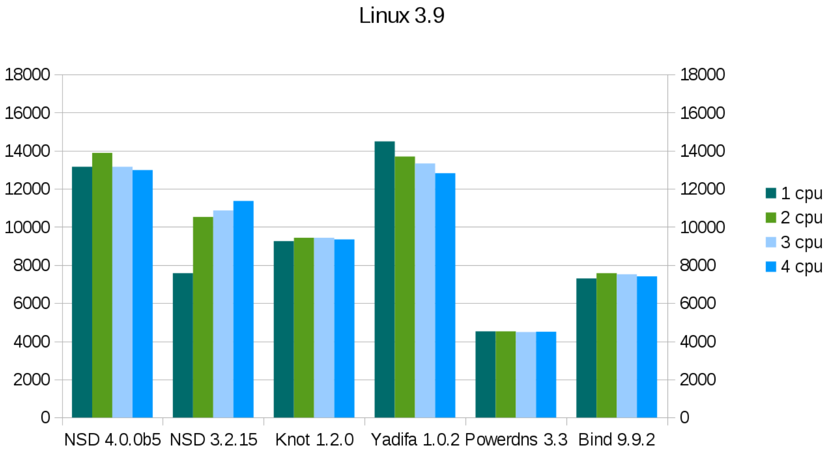
4 cpu

Performance measurements - Results



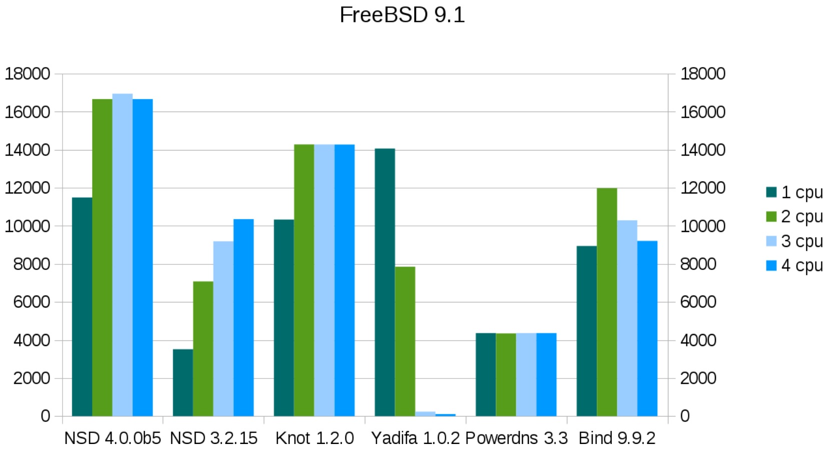
Performance measurements - TCP

- Using PowerDNS's `dnstcpbench`
- queries per second (tenfold slower)



Performance measurements - TCP

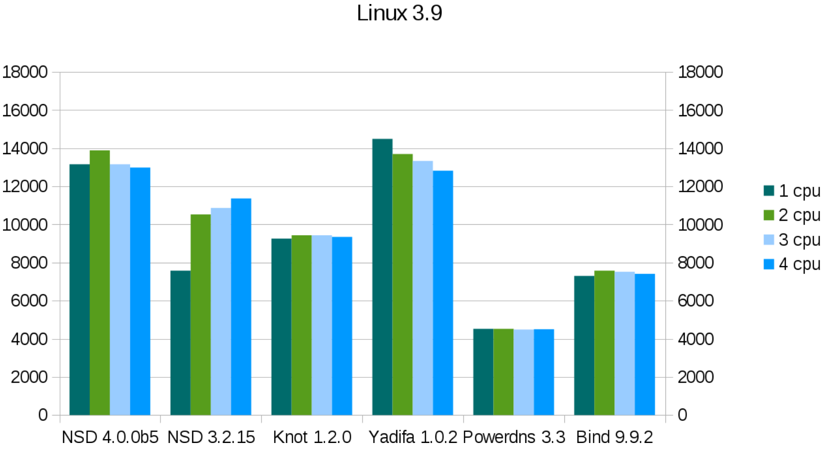
- Using PowerDNS's `dnstcpbench`
- queries per second (tenfold slower)
- FreeBSD sends connection resets when out of backlog (counted as qps, but compensated)



Performance measurements - TCP

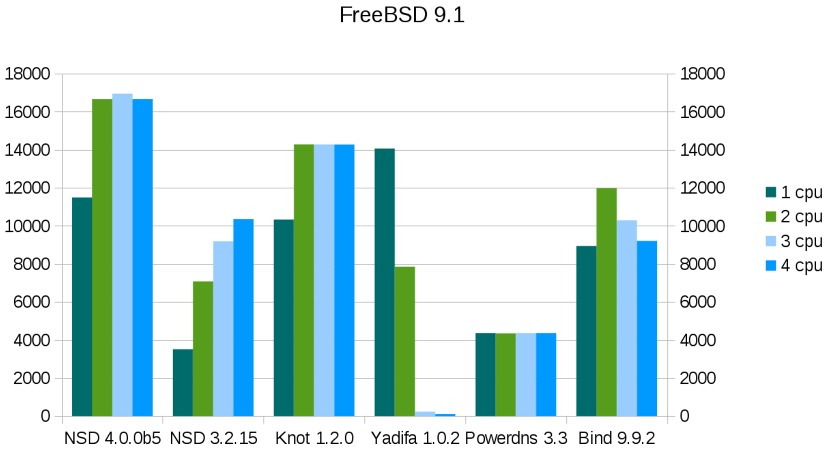
- Because degrades are slight, maintain UDP advise

Bind & Knot	Use 4 out of 4 cores	(use all cores?)
Yadifa	Use 3 out of 4 cores	(use # cores - 1?)
NSD	Use 2 out of 4 cores	(use # cores - 2?)



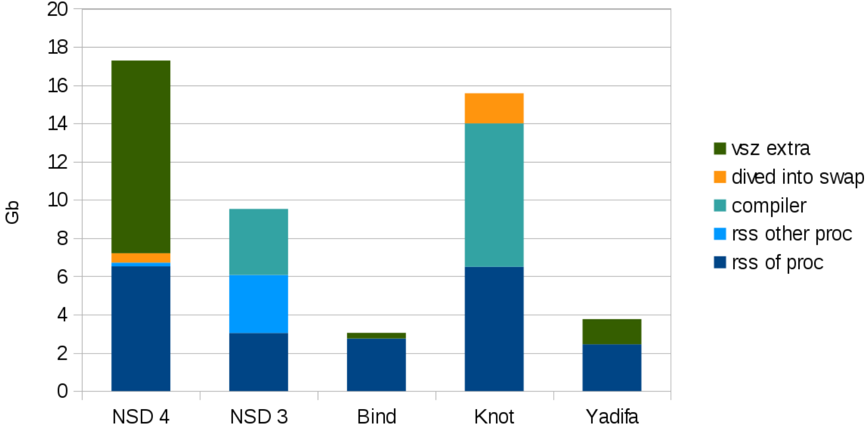
Performance measurements - TCP

- Because degrades are slight, maintain UDP advise
- Same holds for FreeBSD: Use all 4 cores (except with Yadifa)



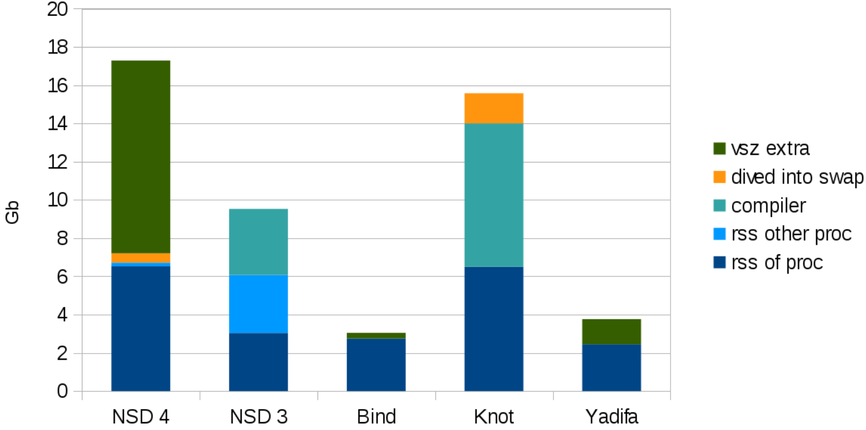
Memory usage

- Same machine as before (8GB Ram)
- All name servers loaded with the .nl zone of June 2013
  (1.5GB, 5.3 million delegations, NSEC3 opt-out, 28% signed delegations)



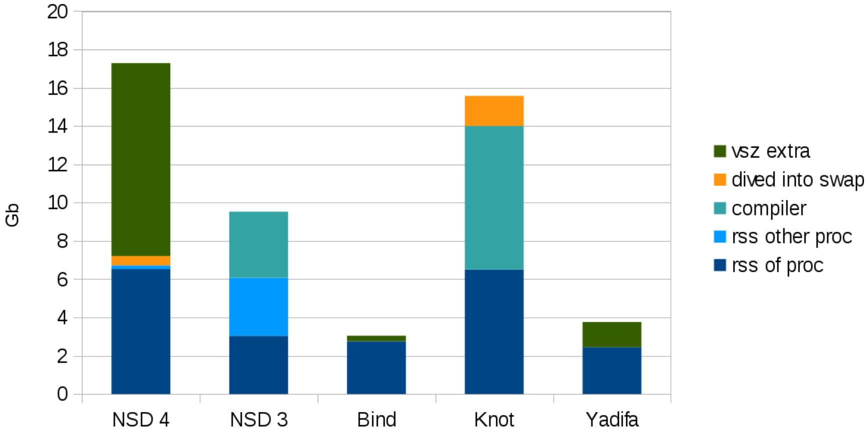
Memory usage

- Same machine as before (8GB Ram)
- All name servers loaded with the .nl zone of June 2013
  (1.5GB, 5.3 million delegations, NSEC3 opt-out, 28% signed delegations)
- Bind and Yadifa easily operate within 4GB



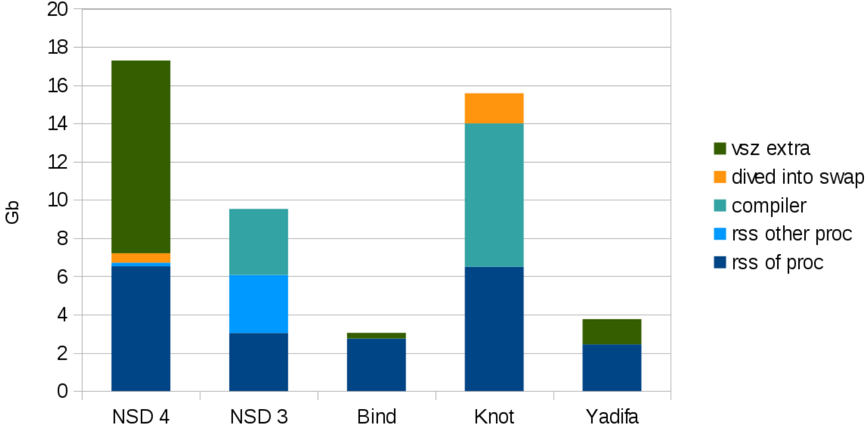
Memory usage

- Bind and Yadifa easily operate within 4GB
- Zone compilation may be performed elsewhere



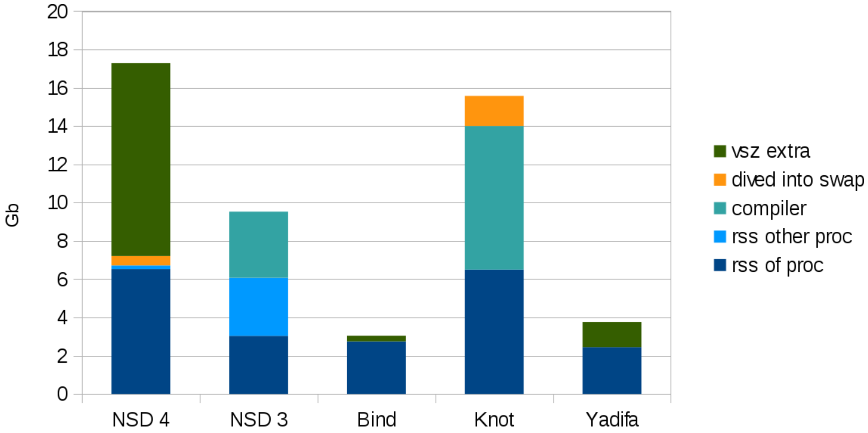
Memory usage

- Bind and Yadifa easily operate within 4GB
- Zone compilation may be performed elsewhere
- But NSD3 needs another rss chunk (3GB) for nsd-patch
  (separate process rewriting back-end and writing out slave zone files)



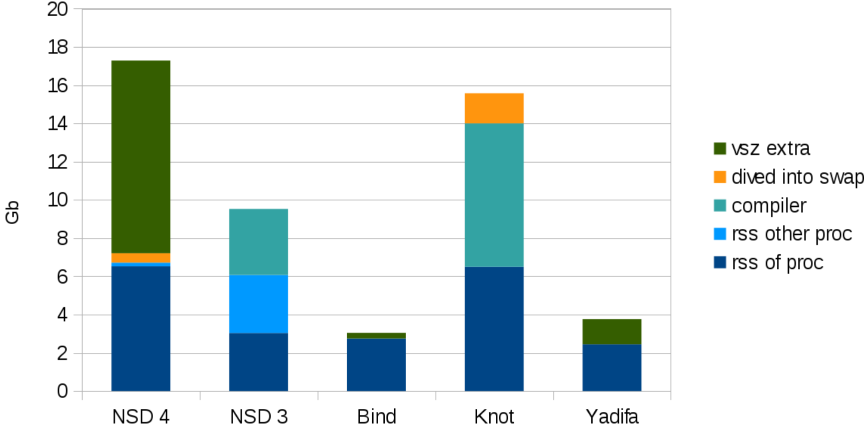
Memory usage

- VSZ Memory of NSD4 is the “mmapped” back-end



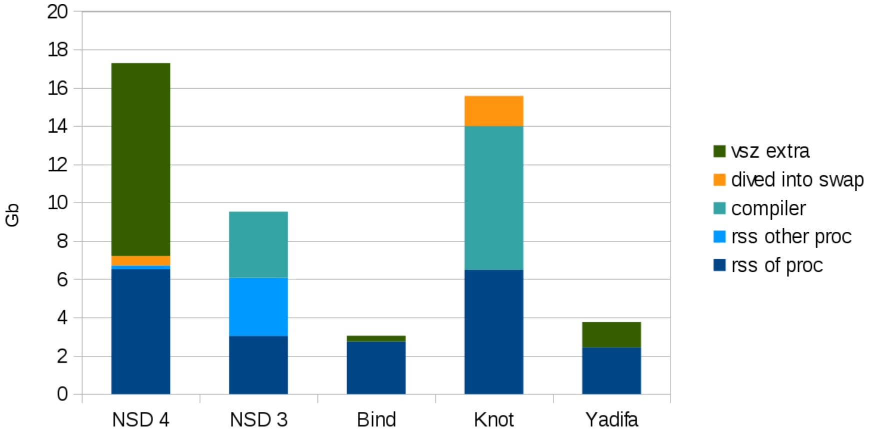
Memory usage

- VSZ Memory of NSD4 is the “mmapped” back-end
- NSD4 needs another rss chunk (6GB)
  For complete zone updates (AXFR or zone file changes)



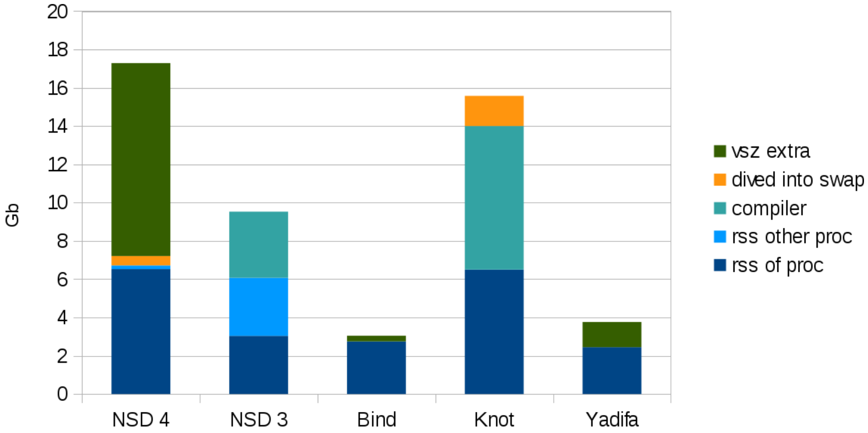
Memory usage

- VSZ Memory of NSD4 is the “mmapped” back-end
- NSD4 needs another rss chunk (6GB)
- But as a slave with only IXFR it just works



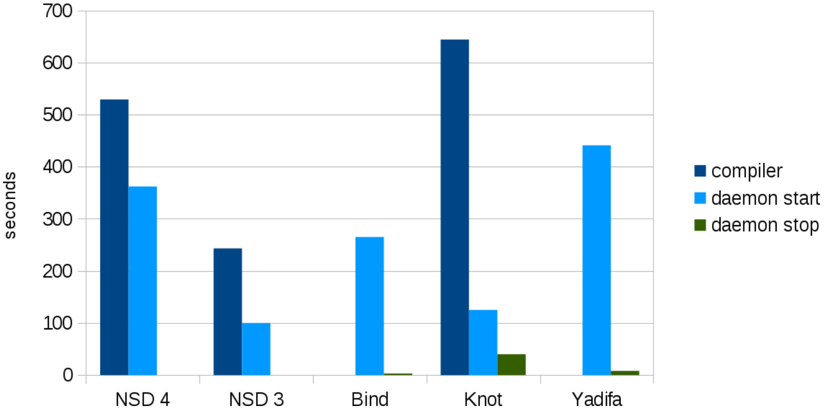
Memory usage

- VSZ Memory of NSD4 is the "mmapped" back-end
- NSD4 needs another rss chunk (6GB)
- But as a slave with only IXFR it just works
- But between 9GB and 17GB Ram would be much better



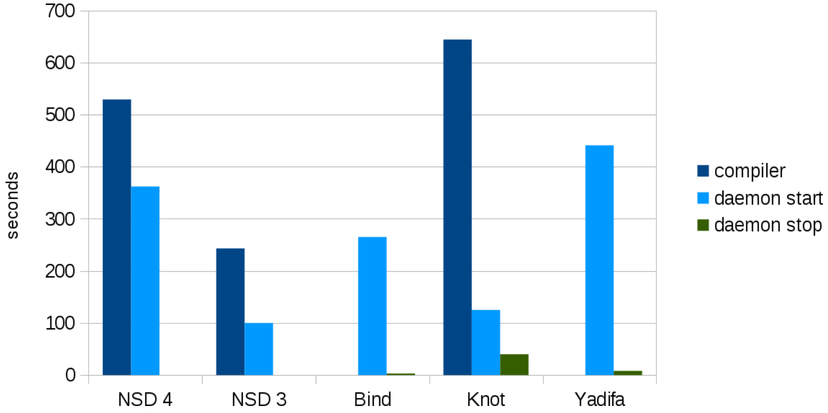
Starting / Stopping

- Knot without the ragel zone parser
- also, Knot zone compiler ran into swap space



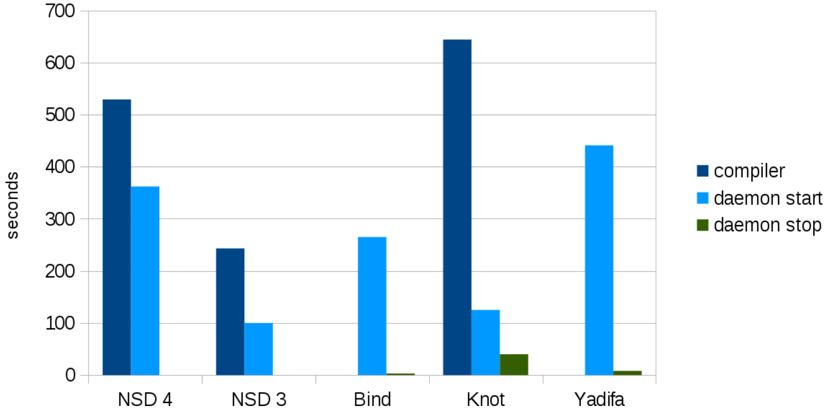
Starting / Stopping

- Knot without the ragel zone parser
- also, Knot zone compiler ran into swap space
- NSD4 zone compiler writes the “mmapped” back-end



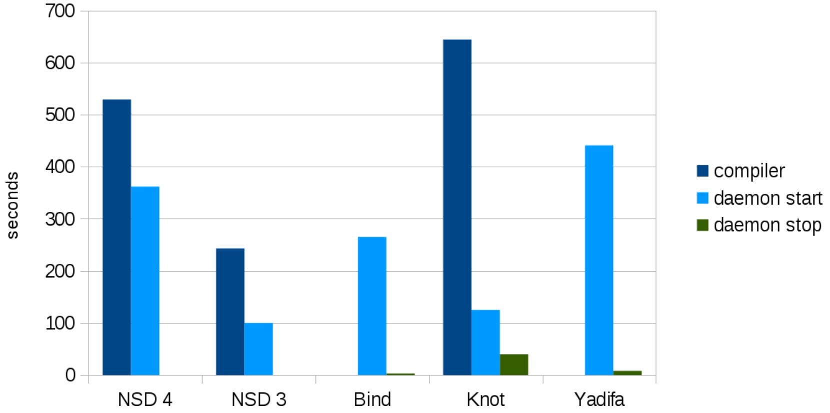
Starting / Stopping

- Do you need large updated/new zone files?



Starting / Stopping

- Do you need large updated/new zone files?
- NSD3 and NSD4 stop quickly (no updates to write out)



Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux (except NSD3 and Yadifa on TCP)

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server

On Linux:

Bind & Knot	Use 4 out of 4 cores	(use all cores?)
Yadifa	Use 3 out of 4 cores	(use # cores - 1?)
NSD	Use 2 out of 4 cores	(use # cores - 2?)

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server
- Memory
  - Bind and Yadifa use the least memory

Overall Observations

With our test set up

- Speed
    - FreeBSD is faster than Linux
    - On FreeBSD CPU cores are more in service of the name server
- Memory
    - Bind and Yadifa use the least memory
    - NSD4 memory requirements depend on the size of updates

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server
- Memory
  - Bind and Yadifa use the least memory
  - NSD4 memory requirements depend on the size of updates
- Manageability
  - Do you have large updated/new zone files?

    yes Bind and Yadifa start up quickest

    no NSD3 and Knot start up quickest

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server
- Memory
  - Bind and Yadifa use the least memory
  - NSD4 memory requirements depend on the size of updates
- Manageability
  - Do you have large updated/new zone files?

    yes Bind and Yadifa start up quickest

    no NSD3 and Knot start up quickest
  - NSD3 and NSD4 stop quickest (more crash resistant?)

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server
- Memory
  - Bind and Yadifa use the least memory
  - NSD4 memory requirements depend on the size of updates
- Manageability
  - Do you have large updated/new zone files?

    yes Bind and Yadifa start up quickest

    no NSD3 and Knot start up quickest
  - NSD3 and NSD4 stop quickest
  - Only NSD3 needs restart for new zones (but starts quickest)

Overall Observations

With our test set up

- Speed
  - FreeBSD is faster than Linux
  - On FreeBSD CPU cores are more in service of the name server
- Memory
  - Bind and Yadifa use the least memory
  - NSD4 memory requirements depend on the size of updates
- Manageability
  - Do you have large updated/new zone files?
  - NSD3 and NSD4 stop quickest
  - Only NSD3 needs restart for new zones

but...

- Test was to measure and compare NSD4
- Need to test different processor types (quad-cores)
- Need to test different network cards (intel)

Colophon

All measurements were performed in June by Wouter Wijngaards

Blog posts

NSD4 Performance Measurements

http://nlnetlabs.nl/blog/nsd4-performance-measurements/

NSD4 High Memory Usage

http://nlnetlabs.nl/blog/nsd-4-mem-use/

NSD4 TCP Performance

http://nlnetlabs.nl/blog/nsd4-tcp-performance/