Domain Name Server Comparison:

BIND 8 vs. BIND 9 vs. djbdns vs. ???

Brad Knowles

Senior Consultant for Snow, BV

brad.knowles@snow.nl

http://www.shub-internet.org/brad/papers/dnscomparison/

Overview

- Meta Information
- TLD Survey Results
- Software
  - Installation
  - Features
  - Performance
- Conclusions

Meta Information

- Hardware Used
- Software Used
- Methodology

Hardware Used

- TLD Survey
  - OS: BSD/OS 4.2
  - CPU: Pentium III
  - RAM: 512MB real, 1.0GB virtual

Hardware Used

- Performance Testing
  - Compaq Armada 4131T Laptop
    - OS: FreeBSD 4.6.2-RELEASE
    - CPU: Pentium 133
    - RAM: 48MB real, 384MB virtual
    - NICs: Asanté FriendlyNET AL1011
      "Prism2" 802.11b WiFi PC Card
      & Linksys EtherFast 10/100 PC Card (PCM100)
    - HD: 10GB IBM Travelstar 20GN
      - 4200 RPM
      - 12ms avg. seek

Hardware Used:

Performance Testing

Software Used

- ISC
  - BIND 8.3.3-REL
  - BIND 9.2.2rc1
- djbdns 1.05
  - daemontools 0.76
  - ucpsi-tcp 0.88
  - tinydns-bent 1.1
- nsd 1.02b1
- Nominum
  - ANS (Authoritative Name Server) 2.0.1-1eval
  - CNS (Caching Name Server) 1.1.0b1
- PowerDNS 2.9.4

Some Software Considered

- QuickDNS (authoritative)
  - See <http://www.menandmice.com/2000/2600_isp_dns_solution.html>
    - Aimed at small-to-medium size businesses, ISP/ASPs, Enterprise customers, full AD integration, management interface, debugging & wizard tools (including DNS Expert Monitor), managed service available, integrates with QuickDNS server as well as stock ISC BIND, **not yet available for testing platform**
- MaraDNS (authoritative & caching)
  - See <http://www.maradns.org/>
- Pdnsd (caching)
  - See <http://home.t-online.de/home/Moestl/>
- Posadis (authoritative)
  - See <http://posadis.sourceforge.net/>
- MyDNS (authoritative)
  - See <http://mydns.bboy.net/>
    - Front-end to MySQL

Some Not Considered

- LDAPDNS
  - See <http://www.nimh.org/code/ldapdns/>
    - Hacked version of djbdns on top of OpenLDAP, too buggy
- UltraDNS
  - See <http://www.ultradns.com/>
    - Managed service, not software
- Cisco Network Registrar
  - See <http://www.cisco.com/warp/public/cc/pd/nemnsw/nerr/prodlit/cnr30_ov.htm>
    - Aimed primarily at Enterprise and broadband clients with management interface, integrates many non-DNS related features, **not available for test platform**
- Incognito DNS Commander
  - See <http://www.incognito.com/products/DNSCommander/Enterprise/index.asp>
    - Aimed primarily at Enterprise and broadband clients with management interface or managed service, **not available for test platform**

Methodology

- TLD Survey
  - Synthesize root zone
    - Why?
    - Root itself & root nameservers
    - Original gTLDs
      - arpa, com, edu, gov, int, mil, net, org
    - New gTLDs (http://www.icann.org/tlds/)
      - aero, biz, coop, info, museum, name, pro
    - ccTLDs
      - Try “aa” through “zz”

Methodology

- Query all zones
- Query all detected nameservers for all valid zones
- Check for
  - What software?
  - Open Recursive/Caching?
    - Among other things. See the LISA 2002 version of these slides at <http://www.shub-internet.org/brad/papers/dnscomparison/> for some others.

Methodology

- Performance Testing
  - Build from source or packages, according to instructions
  - Test caching & authoritative (as appropriate)
  - Test with root & “tv” zones
    - Root is well-known small zone
    - Largest zone I could get was “tv”, ~20MB
  - Tests generated on local machine
    - Use loopback network for all queries
    - Should avoid most questions of NIC or network configuration & performance
    - Each run lasts sixty seconds, then sleep for ten

Methodology

- Do multiple test runs (usually 100)
  - Helps smooth out inter-test variations
  - Helps ensure that we’re testing the local software and not the speed of my Internet connection
- Test input
  - Take all labels from records within zone
  - Convert to lowercase, sort, & uniq
  - For each label
    - Generate queries for SOA, NS, MX, and A (in that order)
  - Responses will be mixture of NXDOMAIN and data
    - We don’t care, we just want to get an answer (any answer is okay) and not generate out-of-zone questions
      - » Otherwise, *djbdns/tinydns* might ignore us

Methodology

– Sample test input for the root zone

```
AC        SOA
AC        NS
AC        MX
AC        A
AD        SOA
AD        NS
AD        MX
AD        A
AE        SOA
AE        NS
:
:
```

Methodology

- Nominum white paper “How to measure the Performance of a Caching DNS Server”
    - See <http://www.nominum.com/content/documents/CNS_WP.pdf>
  - Latency measurements should be made by single-threading queries
  - Capture snapshot of real-world traffic in your production network for use in simulation, or test on real-world network
  - If you test in the lab, use one with replica of Internet zones on multiple servers, fast enough to ensure that they can’t be the bottleneck
  - If synthetic, test input should be randomized
  - For throughput testing, *queryperf* was designed to test low-latency authoritative servers, not high-latency caching servers
    - Make sure to increase the number of outstanding queries
    - Use multiple query sources to generate enough traffic

Methodology

- Very useful & interesting paper — But …
  - What tools exist to measure nameserver latency?
  - I’m not here to measure latency. My ISDN line is too slow. I want to measure authoritative & caching efficiency and relative performance on the same hardware & network.
  - The real-world traffic load on my home network is minimal. “Snapshot” capture & replay is not practical for me. I have no choice but to create a synthetic test suite.
  - QUERY_LOG is one of the most expensive things you can turn on for any nameserver. If people can turn on QUERY_LOG, they don’t need to worry about whether they needed to upgrade their servers or change software for performance.
    - If not QUERY_LOG, then what tools can you use to capture a “snapshot”?
    - Regardless of how you capture the “snapshot”, how do you replay it? How do you replay it with millisecond accuracy?

Methodology

- Very useful & interesting paper — But …
  - My test lab is one machine, two at the most. I don’t have the resources available to create a sophisticated closed lab, so I have no choice but to test on the “live Internet”.
    - RIPE NCC/RIPE DNS WG/DISTEL to the rescue?
  - Query randomization to avoid extreme OS pre-caching (e.g., doing an ordered scan of an indexed database) is a good idea, but so far as I know, no such tools exist.
    - Should *queryperf* be modified to perform input randomization?
    - Can someone at least write a decent Perl script for this?
  - On my ultra low-powered system, *queryperf* appeared to do just fine testing caching nameservers. However, other testing results I’ve recently received lead me to believe I should re-run all of my tests with higher numbers of *queryperf* threads, just to be sure.

TLD Survey

- Zone Information
- What Software?
- Open Recursive/Caching?
- Risks

TLD Survey

- Total 257 zones:
  - Root (.)
  - gTLDs
    - arpa com edu gov int mil net org aero biz coop info museum name pro
  - ccTLDs
    - ac ad ae af ag ai al am an ao aq ar as at au aw az ba bb bd be bf bg bh bi bj bm bn bo br bs bt bv bw by bz ca cc cd cf cg ch ci ck cl cm cn co cr cu cv cx cy cz de dj dk dm do dz ec ee eg er es fi fj fk fm fo fr ga gb gd ge gf gg gh gi gl gm gn gp gq gr gs gt gu gw gy hk hm hn hr ht hu id ie il im in io iq ir is it je jm jo jp ke kg kh ki km kn kr kw ky kz la lb lc li lk lr ls lt lu lv ly ma mc md mg mh mk ml mm mn mo mp mq mr ms mt mu mv mw mx my mz na nc ne nf ng ni nl no np nr nu nz om pa pe pf pg ph pk pl pm pn pr ps pt pw py qa re ro ru rw sa sb sc se sg sh si sj sk sl sm sn so sr st su sv sy sz tc td tf tg th tj tk tm tn to tp tr tt tv tw tz ua ug uk um us uy uz va vc ve vg vi vn vu wf ws ye yt yu za zm zw

TLD Survey

- Total 742 unique server/IP pairs
- Top Ten:

86	ns.ripe.net	193.0.0.193
45	ns.uu.net	137.39.1.3
40	sunic.sunet.se	192.36.125.2
39	ns.eu.net	192.16.202.11
29	munnari.oz.au	128.250.1.21
25	auth02.ns.uu.net	198.6.1.82
22	rip.psg.com	147.28.0.39
18	ns-ext.vix.com	204.152.184.64
17	ns2.nic.fr	192.93.0.4
9	dns.princeton.edu	128.112.129.15

TLD Survey:
What Software?

- Methodology
  - Query #1:

```
dig @server chaos txt version.bind
```

  - Query #2:

```
dig @server chaos txt authors.bind
```

  - Sift through responses to try to classify versions

TLD Survey:
What Software?

- Decision Tree:
  - Responds to both queries => BIND 9
    - Including “REFUSED” & “NXDOMAIN”
  - Responds to first query only => BIND 4.9.3 - 8
    - Including “REFUSED” & “NXDOMAIN”
    - Responds with “SERVFAIL” for second query
  - Responds with “NOTIMPL” => BIND < 4.9.2 & NT4/Win2k DNS
  - Responds with “FORMERR” => tinydns 1.05
  - No response at all => tinydns < 1.05 or network problem

TLD Survey:

What Software?

- Claimed Version Top Ten

85 “9.2.1”	34 SERVFAIL
84 “8.2.3-REL”	22 “8.2.2-P5”
58 “8.3.3-REL”	20 “9.2.0”
45 timed out	20 “8.3.1-REL”
41 REFUSED19	“8.2.5-REL”

TLD Survey:
What Software?

- Fingerprint analysis

415	BIND-4.9.3+/8	55.93%
252	BIND-9	33.96%
34	SERVFAIL	4.58%
20	UltraDNS	2.70%
7	BIND-4 (<4.9.2 or NT/Microsoft DNS)	0.94%
6	TIMEOUT	0.81%
3	PowerDNS	0.40%
3	Incognito	0.40%
2	djbdns/tinydns-1.05	0.27%

TLD Survey:

What Software?

- Root & gTLDs (arpa, com, edu, gov, mil, org)
  - BIND-8
- int, museum, name, pro
  - BIND-8 & BIND-9

TLD Survey:
What Software?

- aero
  - BIND-9 & UltraDNS
- biz
  - SERVFAIL
- coop, info
  - UltraDNS

TLD Survey:
What Software?

- Interesting Versions for ccTLDs
  - UltraDNS (cx, ie, lu, no)
    - Zones also served by BIND-8 & BIND-9
  - Incognito DNS Commander (aq, pn)
    - Zones also served by BIND-9
  - PowerDNS (tk)
  - tinydns-1.05
    - ns-soa.darenet.dk (dk, gl)
    - ns3.utoronto.ca (ca)
      - Zones also served with BIND-8 & BIND-9

TLD Survey:
Open Recursive/Caching?

- Methodology
  - For each zone & server, do
    - Query server for obvious out-of-zone data with recursion off
    - Repeat query with recursion on
    - Repeat query with recursion off again
      - If 1st response is referral, and 2nd and 3rd responses have the “ra” bit set (and are the same, modulo TTL differences), then the server is open recursive/caching

TLD Survey:

Open Recursive/Caching?

- Example

```
dig @server thisisan.obviousnonexistentdomain.com. any +norec
dig @server thisisan.obviousnonexistentdomain.com. any +rec
dig @server thisisan.obviousnonexistentdomain.com. any +norec
```

TLD Survey:
Open Recursive/Caching?

- 204 zones have one or more recursive/caching servers
  - 79.3% of all root & TLD zones are affected
  - gTLDs
    - aero museum
  - ccTLDs
    - ac ad ae ag ai al am an ar as au aw az ba bb bd bf bg bh bi bj bm bn bo bs bt bv bw by ca cd cf cg ch ci ck cl cm cn co cr cu cy dj dk do dz ec ee eg er es fi fj fk fm fo fr ga gb gd gf gg gh gi gl gm gn gp gr gs gt gu gw gy hk hn hr ht hu id il im in int io iq ir it je jm jo jp ke kg kh ki km kn kw kz la lb lc li lk lr ls lt lu lv ma mc md mg mh mk ml mm mn mo mp mq mr ms mt mu mv mw my mz na nc ne nf ng ni no np nr nz om pa pe pf pg pk pl pr py qa ro ru rw sa sb sc se sg sh si sj sk sl sm sn so sr st su sv sy sz tc tf tg th tj tm tn to tp tr tt tz ua ug uk um uy uz va ve vg vi vn vu ws yu za zm zw

TLD Survey:

Open Recursive/Caching?

- 398 Servers are affected
  - 53.6% of all root & TLD servers
- Top Ten

22	rip.psg.com	8	upr1.upr.clu.edu
12	ns2.berkeley.edu	8	hippo.ru.ac.za
12	ns1.berkeley.edu	6	ns.ird.fr
12	ns0.ja.net	5	joanna.william.org
9	merapi.switch.ch	5	f.i-dns.net

TLD Survey:

Risks

- Open Recursive/Caching Server
  - Pro
    - Others may not have access to good recursive/caching name service
    - Done by default, easier to leave turned on

TLD Survey:

Risks

- Open Recursive/Caching Server, Cons
  - Anyone can abuse your server
    - Including using your server to DoS another
    - Including having your server effectively host their domain
  - Leaves you much more vulnerable to cache poisoning/pollution
    - If you are also authoritative, you risk passing on poison/pollution to unsuspecting clients
      - Legal liabilities?
    - Hostname-based security can be easily by-passed
    - Poisoned/polluted parent zone increases security risk for all children
    - Eugene Kashpureff used this attack in 1997
  - Causes one server (or set of servers) to do much more work than would otherwise be necessary

TLD Survey:

Risks

- Open Recursive/Caching Server, Conclusion
  - Generally speaking, you should split functions onto separate machines or IP addresses
    - Authoritative servers should be authoritative-only
      - Also disable “fetch-glue”
    - Recursive/caching servers should not be authoritative
  - Recursive/caching servers should only answer queries from “internal” sources

Software

- Installation & Configuration
- Features
- Performance

Software:

Installation & Configuration

- ISC
  - BIND-8
  - BIND-9
- djbdns
- nsd
- Nominum
  - CNS
  - ANS
- PowerDNS

Software:

Installation

- BIND-8
  - ftp ftp://ftp.isc.org/isc/bind/src/cur/bind-8/* .
  - Verify checksums

    ```
    gtar zxf bind-src.tar.gz
    cd src
    make depend
    make all
    make install
    ```

  - Create `/etc/named.conf` & zone files

Software: Configuration

- BIND-8 `/etc/named.conf`

```
options {
        directory "/var/named";
};

zone "." IN {
        type hint;
        file "named.ca";
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
};
```

Software: Configuration

- BIND-8 `/var/named/named.ca`

```
.                          3600000  IN  NS    A.ROOT-SERVERS.NET.
.                          3600000      NS    B.ROOT-SERVERS.NET.
.                          3600000      NS    C.ROOT-SERVERS.NET.
.                          3600000      NS    D.ROOT-SERVERS.NET.
.                          3600000      NS    E.ROOT-SERVERS.NET.
.                          3600000      NS    F.ROOT-SERVERS.NET.
.                          3600000      NS    G.ROOT-SERVERS.NET.
.                          3600000      NS    H.ROOT-SERVERS.NET.
.                          3600000      NS    I.ROOT-SERVERS.NET.
.                          3600000      NS    J.ROOT-SERVERS.NET.
.                          3600000      NS    K.ROOT-SERVERS.NET.
.                          3600000      NS    L.ROOT-SERVERS.NET.
.                          3600000      NS    M.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.        3600000      A     198.41.0.4
B.ROOT-SERVERS.NET.        3600000      A     128.9.0.107
C.ROOT-SERVERS.NET.        3600000      A     192.33.4.12
D.ROOT-SERVERS.NET.        3600000      A     128.8.10.90
E.ROOT-SERVERS.NET.        3600000      A     192.203.230.10
F.ROOT-SERVERS.NET.        3600000      A     192.5.5.241
G.ROOT-SERVERS.NET.        3600000      A     192.112.36.4
H.ROOT-SERVERS.NET.        3600000      A     128.63.2.53
I.ROOT-SERVERS.NET.        3600000      A     192.36.148.17
J.ROOT-SERVERS.NET.        3600000      A     198.41.0.10
K.ROOT-SERVERS.NET.        3600000      A     193.0.14.129
L.ROOT-SERVERS.NET.        3600000      A     198.32.64.12
M.ROOT-SERVERS.NET.        3600000      A     202.12.27.33
```

Software:
Configuration

- BIND-8 `/var/named/localhost.zone`

```
$TTL    86400
$ORIGIN localhost.
@               1D IN SOA       @ root (
                      42      ; Serial (D. Adams)
                      3H      ; Refresh
                      15M     ; Retry
                      1W      ; Expiry
                      1D )    ; NegCache TTL

                1D IN NS      @
                1D IN A       127.0.0.1
```

Software:

Configuration

- BIND-8 `/var/named/named.local`

```
$TTL     86400
@       IN SOA localhost. root.localhost.  (
                1997022700    ; Serial YYYYMMDDNN
                28800         ; Refresh
                14400         ; Retry
                3600000       ; Expire
                86400 )       ; NegCache TTL
        IN NS  localhost.

1       IN PTR localhost.
```

Software:

Installation

- BIND-9
  - ftp `ftp://ftp.isc.org/isc/bind9/9.2.2rc1/* .`
  - Verify checksums

    ```
    gtar zxf bind-9.2.2rc1.tar.gz
    cd bind-9.2.2rc1
    ./configure
    make
    make install
    rndc-confgen -a
    ```
  - Create `/etc/named.conf` & zone files
    - No changes from BIND-8

Software:

Installation

- djbdns (`tinydns/dnscache`)
  - daemontools

```
mkdir -p /package
chmod 1755 /package
cd /package
wget http://cr.yp.to/daemontools/daemontools-0.76.tar.gz
gtar zxpf daemontools-0.76.tar.gz
cd admin/daemontools-0.76
package/install
```

- On BSD systems, reboot to start `svscan`

Software:

Installation

- djbdns (tinydns/dnscache)
  - ucspi-tcp

```
wget http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz
gtar zxf ucspi-tcp-0.88.tar.gz
cd ucspi-tcp-0.88
make
make setup check
```

Software:

Installation

- djbdns (`tinydns/dnscache`)
  - djbdns proper

```
wget http://cr.yp.to/djbdns/djbdns-1.05.tar.gz
gtar zxf djbdns-1.05.tar.gz
cd djbdns-1.05
make
make setup check
```

  - Documentation

```
wget http://cr.yp.to/djbdns/doc.tar.gz
gzcat doc.tar.gz | (cd /; tar -xf -)
wget http://cr.yp.to/slashdoc/slashdoc-merge
./slashdoc-merge
```

Software:
Configuration

- djbdns (`tinydns/dnscache`)
  - Local-only `dnscache`
    - As root
      - Create accounts “`Gdnscache`” and “`Gdnslog`”
      - Create `/etc/dnscache` service directory
      - Run the commands:

        ```
        dnscache-conf Gdnscache Gdnslog /etc/dnscache
        ln -s /etc/dnscache /service
        sleep 5
        svstat /service/dnscache
        ```

      - In your `/etc/resolv.conf`, put:

        ```
        nameserver 127.0.0.1
        ```

Software: Configuration

- djbdns (`tinydns/dnscache`)
  - Network `dnscache`
    - As root
      - Create accounts “`Gdnscache`” and “`Gdnslog`”
      - Create `/etc/dnscache` service directory
      - Run the commands:

        ```
        dnscache-conf Gdnscache Gdnslog /etc/dnscache \
        10.53.0.1
        ln -s /etc/dnscache /service
        sleep 5
        svstat /service/dnscache
        touch /etc/dnscache/root/ip/10
        ```

      - In your `/etc/resolv.conf`, put:

        ```
        nameserver 10.53.0.1
        ```

Software: Configuration

- djbdns (`tinydns/dnscache`)
  - `Tinydns` (UDP & TCP, no zone transfers)
    - As root
      - Create accounts “`Gtinydns`”, “`Gaxfrdns`” and “`Gdnslog`”
      - Create `/etc/tinydns` and `/etc/axfrdns` service directories
      - Run the commands:

```
tinydns-conf Gtinydns Gdnslog /etc/tinydns 192.168.0.5
axfrdns-conf Gaxfrdns Gdnslog /etc/axfrdns \
/etc/tinydns 192.168.0.5
echo ':allow,AFXR=""' > /etc/axfrdns/tcp
cd /etc/axfrdns
make
ln -s /etc/tinydns /service
ln -s /etc/axfrdns /service
sleep 5
svstat /service/tinydns
svstat /service/axfrdns
```

      - Update `/service/tinydns/root/data` and then run the command:

```
make
```

Software:
Configuration

- djbdns (`tinydns/dnscache`)
  - Sample `/service/tinydns/root/data` format

```
# Delegated nameserver records (someone else provides the SOA)
#
#   &fqdn:ip:x:ttl =>
#
#       NS record x.ns.fqdn as nameserver for fqdn
#       A record mapping x.ns.fqdn -> ip [if ip present]
&.::a.root-servers.net.:518400
&.::b.root-servers.net.:518400
&.::c.root-servers.net.:518400
&.::d.root-servers.net.:518400
&.::e.root-servers.net.:518400
&.::f.root-servers.net.:518400
&.::g.root-servers.net.:518400
&.::h.root-servers.net.:518400
&.::i.root-servers.net.:518400
&.::j.root-servers.net.:518400
&.::k.root-servers.net.:518400
&.::l.root-servers.net.:518400
&.::m.root-servers.net.:518400
```

Software: Configuration

- djbdns (tinydns/dnscache)
  - Sample /service/tinydns/root/data format

```
# Zone records
#
#   Zfqdn:ns:contact:serial:refresh:retry:expire:minimum:ttl =>
#
#       SOA record giving ns as primary nameserver for fqdn
#       all options can be expressed just as they occur
#       in a zone file; e.g. contact is user.fqdn; the
#       first . must be replaced by @ to produce email addr
Z.:a.root-servers.net.:nstld.verisign-
grs.com.:2002101601:1800:900:604800:86400:86400
# Service records (host aliases)
#
#   +fqdn:ip:ttl =>
#
#       A record mapping fqdn -> ip
+uucp-gw-2.pa.dec.com:16.1.0.19:172800
+ns2.psi.net:38.8.50.2:172800
+ns5.jaring.my:61.6.38.139:172800
```

Software: Configuration

- djbdns (`tinydns/dnscache`)
  - Sample `/service/tinydns/root/data` format

```
# TXT records
#
#   'fqdn:s:ttl =>
#
#       TXT record for fqdn with data s (octal escapes work)
'vrsn-end-of-zone-marker-dummy-record.root:plenus:172800
# MX records (mail exchange)
#
#   @fqdn:ip:x:dist:ttl =>
#
#       MX record showing x.mx.fqdn as mail exchanger for fqdn
at
#               distance dist
#       A record mapping fqdn -> ip
@ww.tv::nomail.www.tv.:10:7200
@wwww.tv::nomail.www.tv.:10:7200
```

Software: Configuration

- djbdns (`tinydns/dnscache`)
  - Sample `/service/tinydns/root/data` format

```
# CNAME records
#
#   Cfqdn:realname:ttl =>
#
#       CNAME record for fqdn pointing to domain name realname
Cnx--1a000028787fj.tv:ra--gbfeuvkl.tv.:7200
Cnx--1a002drdrfmfmbayd.tv:ra--gbfjgtcp.tv.:7200
Cnx--1a002fefefvfvfe.tv:ra--gbfeiv2e.tv.:7200
```

Software:

Installation

- nsd (Name Server Daemon)
  - Create “`nsd`” user (and optionally, “`nsd`” group)

    ```
    wget http://www.nlnetlabs.nl/downloads/nsd/nsd-1.0.2b1.tar.gz
    ```

    ```
    gtar zxf nsd-1.0.2b1.tar.gz
    ```
  - Verify options set correctly in “Makefile”
    - E.g., “-DINET6” for IPv6 support

    ```
    make all
    ```

    ```
    make install
    ```

Software:

Configuration

- nsd (Name Server Daemon)
  - If you compiled with support for TCP-Wrappers, enable AXFR in `hosts.allow`, for example:

    ```
    axfr: ALL : allow
    ```
  - Create `/usr/local/etc/nsd/nsd.zones` (see nsd.zones.sample)
    - Uses BIND master file format
  - Copy zone files under `/usr/local/etc/nsd`, as appropriate
  - Run “`nsdc update`”, if necessary
  - Run “`nsdc rebuild`”
  - Run “`nsdc start`”

Software:

Configuration

- nsd `/usr/local/etc/nsd/nsd.zones`

```
zone     tv   tv.zone
zone     .    root.zone
```

Software:

Installation

- Nominum CNS
  - Get `cns-1.1.0b1-freebsd.tar.gz`

    ```
    gtar zxf cns-1.1.0b1-freebsd.tar.gz
    cd cns-1.1.0b1
    pkg_add nomutils-1.31.0.tgz
    pkg_add cns-1.1.0b1.tgz
    ```

  - Create `/etc/cns.conf`
    - See `cns1.1-beta-v2.pdf`

    ```
    /etc/rc.d/init.d/cns start
    ```

Software:

Configuration

- Nominum CNS `/etc/cns.conf`
  - From `cns1.1-beta-v2.pdf`, comments stripped

```
listen-on 0.0.0.0;
command-channel 127.0.0.1 port 9333 "MySecret==";
view "world" IN {
 preload 1.0.0.127.in-addr.arpa. PTR localhost;
 preload localhost. A 127.0.0.1;
 check-responses yes;
};
```

Software:

Installation

- Nominum ANS
  - Get `ans-2.0.1-1eval-freebsd.tgz`

    ```
    gtar zxf ans-2.0.1-1eval-freebsd.tgz
    cd ans-2.0.1-1eval
    pkg_add nomutils-1.32.0.tgz
    pkg_add ansbdb-2.0.1-1eval.tgz
    pkg_add ans-2.0.1-1eval.tgz
    ```
  - Create `/etc/ans directory` & `/etc/ans.conf`
    - See `ans2.0.pdf`

Software:

Installation

- Nominum ANS
  - Start the server

```
ans -f -c /etc/ans.conf
```

    - Note that `/usr/local/etc/rc.d/ans.sh` is created automatically and will run on reboot
  - Import a master database

```
nom_tell ans update-view import='(master)'
```

    - Watch your spaces and quotes!
  - Load a zone

```
ans_load cc default example.com example.db master
```

Software:

Configuration

- Nominum ANS `/etc/ans.conf`
    - Based on `ans2.0.pdf`

```
directory /etc/ans
listen-on 127.0.0.1
db master berk dbfile=master.db
db tv berk dbfile=tv.db
```

Software:

Installation

- PowerDNS
  - Install PostgreSQL

```
cd /usr/ports/databases/postgresql7
make install
wget <http://downloads.powerdns.com/
releases/pdns-2.9.4.tar.gz>
gtar zxf pdns-2.9.4.tar.gz
cd pdns-2.9.4
./configure --with-modules="gpgsql"
gmake
gmake install
```

Software:

Configuration

- Create the “powerdns” database

```
createdb powerdns
```

  - Note that this step is not documented anywhere on the PowerDNS site!!!

- Start PostgreSQL

```
/usr/local/etc/rc.d/010.psql.sh start
```

- Import the PowerDNS base schema
  - See A.5.2 at <http://doc.powerdns.com/generic-mypgsql-backends.html#AEN2799>
  - Example

```
psql powerdns < schema
```

Software:

Configuration

- Need data
  - Import data from BIND zone files & `/etc/named.conf`
    - Via `zone2sql`
  - Could create data directly in PostgreSQL database
    - Perhaps via PowerAdmin PHP interface

Software:
Configuration

- Unable to complete testing on PowerDNS
  - Unfortunately, `zone2sql` needs more work
    - Can’t correctly convert `named.conf` for root zone
      - Hint zone works okay
      - Data for actual root zone gets mis-labeled with empty zone name, thus causing insertion into table to fail
      - Doesn’t grok all formats of symbolic TTLs
        - » In particular, doesn’t understand that 5w6d16h = 3600000
  - So far, my resulting configuration performs too poorly to attempt to benchmark
    - Not enough time to get working properly
    - Not enough time to try BIND back-end instead
    - Have not even attempted to use XDB back-end

Software:

Features

- ISC
  - BIND-8
  - BIND-9
- djbdns
- nsd
- Nominum
  - CNS
  - ANS
- PowerDNS

Software:

Features

- BIND-8
  - Pro
    - Full recursive/caching & authoritative name server implementation
    - Recursive/caching & authoritative services can share IP address
    - Still somewhat faster than BIND-9
    - Explicitly supports 30 different OSes & OS versions
      - aix32 aix4 aux3 bsdos bsdos2 cygwin darwin decunix freebsd hpux hpux10 hpux9 irix linux lynxos mpe netbsd next openbsd qnx rhapsody sco42 sco50 solaris sunos ultrix unixware20 unixware212 unixware7 winnt

Software:

Features

- BIND-8
  - Con
    - Based on “Legacy” (read: spaghetti) Code
      - Increased risk of security failures due to obscurity of code
    - Does Not Handle IPv6
    - Single-threaded
    - Zone Transfers Handled Externally
      - Uses `fork()/exec()` model
    - Near “End-of-Life”
      - No new features
      - Only major security bug fixes will be implemented
    - If target OS does not have explicit “port” support, need to modify existing port to work

Software:

Features

- BIND-9
  - Pro
    - Full recursive/caching & authoritative name server implementation
    - Recursive/caching & authoritative services can share IP address
    - Ground-up re-write, uses latest secure programming techniques
      - Each procedure or function applies near-paranoid checks to input
    - DNS Security
      - DNSSEC (signed zones)
      - TSIG (signed DNS requests)

Software:

Features

- BIND-9
  - Pro
    - IP version 6
      - Answers DNS queries on IPv6 sockets
      - IPv6 resource records (A6, DNAME, etc.)
      - Bitstring Labels
      - Experimental IPv6 Resolver Library
    - Multiprocessor Support
    - Multi-threading Support
      - Capable of answering queries while loading zones
    - DNS Protocol Enhancements
      - IXFR, DDNS, Notify, EDNS0
      - Improved standards conformance

Software:

Features

- BIND-9
  - Pro
    - Views
      - One server process can provide multiple “views” of the DNS namespace based on the IP address of the source, e.g. an “inside” view to certain clients, and an “outside” view to others.
    - Improved Portability Architecture
    - Handles Zone Transfers Internally
      - Via separate thread
      - No `fork()/exec()` overhead
      - Won’t cause memory thrashing
    - Easy to set up in highly secure mode (some OSes do this by default)
      - Chroot()
      - Non-privileged process

Software:

Features

- BIND-9
  - Pro
    - Through GNU Autoconf, supports wide variety of hardware & OSes (basic POSIX support, ANSI-C compiler, & 64-bit integer type), including:
      - AIX 4.3
      - COMPAQ Tru64 UNIX 4.0D
      - COMPAQ Tru64 UNIX 5 (with IPv6 EAK)
      - FreeBSD 3.4-STABLE, 3.5, 4.0, 4.1
      - HP-UX 11.x, x < 11
      - IRIX64 6.5
      - NetBSD 1.5
      - Red Hat Linux 6.0, 6.1, 6.2, 7.0
      - Solaris 2.6, 7, 8
      - Windows NT/W2K

Software:

Features

- BIND-9
  - Pro
    - Also reported to compile on
      - AIX 5L
      - SuSE Linux 7.0
      - Slackware Linux 7.x, 8.0
      - Red Hat Linux 7.1
      - Debian GNU/Linux 2.2 and 3.0
      - OpenBSD 2.6, 2.8, 2.9
      - UnixWare 7.1.1
      - HP-UX 10.20
      - BSD/OS 4.2
      - OpenUNIX 8
      - Mac OS X 10.1

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Violates RFCs
      - By default, does not support zone transfers
        - » Uses separate optional external program (*axfrdns*)
      - By default, does not provide referrals
        - » Root & TLD nameservers do little else **but** referrals
      - By default, does not support TCP
        - » If a response results in truncation in the “Answer” section, the “TC” (truncated) bit should be set, resulting in re-trying the query with TCP
      - Truncates responses illegally
        - » Does not set the “TC” bit

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Provides strange responses to query types it does not support
      - Violates the “Be liberal in what you accept, conservative in what you generate” principle
    - Without third-party patch, neither *tinydns* nor *dnscache* can listen to more than one IP address
    - Because *tinydns* and *dnscache* are separate programs, you cannot have them both listening to port 53 on the same IP address
      - Therefore, you cannot have both authoritative and recursive services on the same machine, unless you use multiple IP addresses

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Does not, and author’s code **will not**, support new DNS features
      - DNSSEC, TSIG, IXFR, NOTIFY, EDNS0, IPv6, etc…
    - Natively supports very limited set of record types (from http://www.fefe.de/djbdns/#recordtypes)
      - SOA, NS, A, MX, PTR, TXT, CNAME
    - Design appears to be aimed at answering some security issues of older versions of BIND
      - Many of which have been fixed by later releases of BIND 8
      - Obviated by BIND 9

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Code still not quite kosher?
      - Appears to reliably drop a certain small percentage of queries
    - No good tools to convert local BIND configuration & zone files
      - Everything (including *tinydns-bent*) seems to assume you will pull zone transfer using *axfr-get*
    - Limited hardware/OS support
      - Difficult to tell how many servers would “just work” based on make file

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Slow?
      - Peak 500 qps, according to TinyDNS FAQ (http://web.archive.org/web/20011007065901/http://cr.yp.to/djbdns/faq/tinydns.html)
      - Anecdotal reports of higher levels of performance, but not tested in controlled environment and not sufficiently documented to be repeatable
      - Personal Testing
        - » Real-world Internet demonstrated *tinydns* to ~250 queries per second (qps)
        - » Private testing demonstrated *tinydns* to ~340 qps
        - » Private testing demonstrated *dnscache* to ~96 qps

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Cons
    - Slow?
      - Rick Jones (ftp://ftp.cup.hp.com/dist/networking/briefs/)
        - BIND 9 demonstrated to ~12,000 qps
        - BIND 8 demonstrated to ~14,000 qps
        - Nominum CNS demonstrated to ~53,000 qps

Software:

Features

- djbdns (`dnscache/tinydns`)
  - Biggest Drawback

CENSORED

Software:

Features

- nsd (Name Server Daemon)
  - From: <http://www.nlnetlabs.nl/nsd/>
    - Authoritative-only, high-performance, simple, open-source name server
      - Developed under the auspices of NLnet Labs
    - Designed to be used & administered only by experienced personnel
      - Very little hand-holding, easy to shoot yourself in the foot

Software:

Features

  - Specifically avoids implementing any feature not needed/desired for the role of root/TLD nameserver
    - Does not (yet) support EDNS0, TSIG, A6, KEY, DNAME, DNSSEC, and perhaps some other features
    - Does not implement NOTIFY, round-robin, or support classes other than “IN”
    - Does not and will not support IXFR, Dynamic Update, and possibly other features
- Pre-computes all possible questions and all possible answers for the zones it is configured to serve, then generates indexed database to provide mapping
  - Has implications for large zones
- Less suitable as a general-purpose authoritative-only nameserver?

- **Now in production use on ns.eu.net**

Software:

Features

- Nominum
  - Foundation Caching Name Server (CNS), from <http://www.nominum.com/product.php?id=1>
    - Optimizing DNS performance - Foundation CNS, a dedicated caching name server, performs better, in name resolutions per second, than any other name server. Foundation CNS is the only caching name server that offers Response Validation. Supports secure DNS – DNSSEC cryptographic validation.
  - Foundation Authoritative Name Server (ANS), from <http://www.nominum.com/product.php?id=2>
    - Foundation ANS is a carrier class DNS server product. ANS was designed from the start for excellent performance as a dedicated authoritative name server. ANS outperforms any other name server product in query responses and is able to scale to millions of names. Supports the DNSSEC protocols.

Software:

Features

- Nominum CNS & ANS
  - Full-service nameserver programs from the people who wrote BIND 9 (under contract to the ISC)
    - Paul Mockapetris invented the DNS in 1983, now Chief Scientist for Nominum
  - Complete re-implementation, with support for all available DNS protocol features
  - OSes Supported
    - Solaris
    - FreeBSD
    - Red Hat (CNS only)

Software:

Features

- Nominum CNS
  - Also performs Response Validation, protecting clients against resolver library write buffer overflow (CERT Advisory CA-2002-19) and read buffer overflow (item 1297 in the CHANGES file for BIND-8.3.3)
  - Fastest caching nameserver in the world?
    - Demonstrated to handle up to ~53,000 queries per second by Rick Jones, see <ftp://ftp.cup.hp.com/dist/networking/briefs/lp2kr_dns_server_results.txt>
  - Easiest installation & configuration that I've ever seen for a caching nameserver

Software:

Features

- Nominum ANS
  - Currently uses Berkeley DB 4.x as back-end database
    - Other back-end databases to follow?
  - Fastest general-purpose authoritative nameserver in the world?
    - Only slightly slower than *nsd*, according to some test results
    - Provides full authoritative nameserver features, unlike *nsd*
  - Easiest installation & configuration that I’ve ever seen for a authoritative nameserver

Software:

Features

- Nominum ANS
  - Very robust
    - Used to provide authoritative DNS services for Nominum Global Name Service (GNS) clients
      - Including secondary.com
      - Including TLD customers
        - » .info, .ie, .lu, .no, .in-addr.arpa
        - » Sold GNS business to UltraDNS in 2002
      - First distributed anycast DNS service at TLD level?
  - Easy to upgrade from BIND
    - Through `ans_import` tool to convert BIND `/etc/named.conf` files and associated zone files

Software:

Features

- PowerDNS
  - From <http://www.powerdns.com/products/powerdns/index.php>
    - The PowerDNS Nameserver is a modern, advanced and high performance authoritative-only nameserver. It is written from scratch and conforms to all relevant DNS standards documents. Furthermore, PowerDNS interfaces with almost any database.
      - Now open source (see <http://www.powerdns.org>)
    - Commercial support & consulting available
    - PowerDNS Express domain/web hosting services also available
    - Caching/recursive nameserver to be integrated into version 2.9.5 or 2.9.6?

Software:

Features

- PowerDNS
  - Supports DNS queries on UDP & TCP, IPv4 & IPv6 networking, AXFR
  - Record types supported
    - A, AAAA, CNAME, HINFO, MX, NAPTR, NS, PTR, RP, SOA, SRV, & TXT
  - OSes Supported
    - Linux (esp. Debian)
    - FreeBSD
    - Solaris 7+ (8+ required for IPv6 & AAAA records)
    - OpenBSD & MacOS X (under development)
  - Documentation needs work ;-(
    - Primarily for interfaces to back-end databases

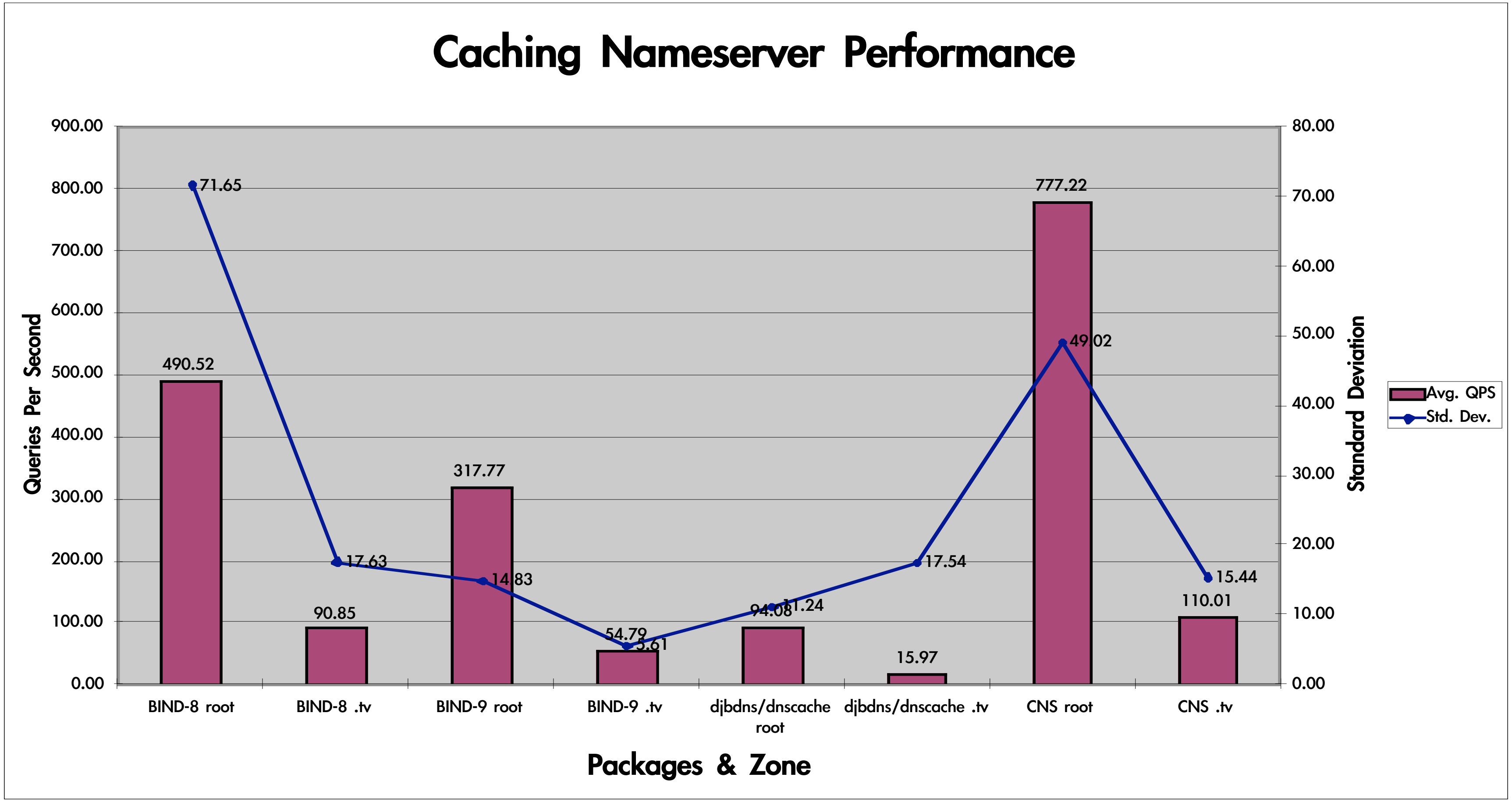
Software:

Features

- PowerDNS
  - Supported back-end methods
    - On Unix/Linux/FreeBSD/etc…
      - MySQL, PostgreSQL, Oracle 8i/9i, IBM DB2
    - On Windows 2000/XP
      - Any ODBC data source including Microsoft SQL Server
    - BIND backend reads `/etc/named.conf` & BIND zone files
      - PowerDNS now 90% drop-in replacement for BIND?
    - Pipe backend for interfacing with arbitrary programs
      - Does not currently work under FreeBSD 4.x/5.x, except as Linux program under Linuxlator
    - XDB backend for special-case situations
      - >50,000 queries per second, as seen for .org
      - Unfortunately, only sparsely documented so far

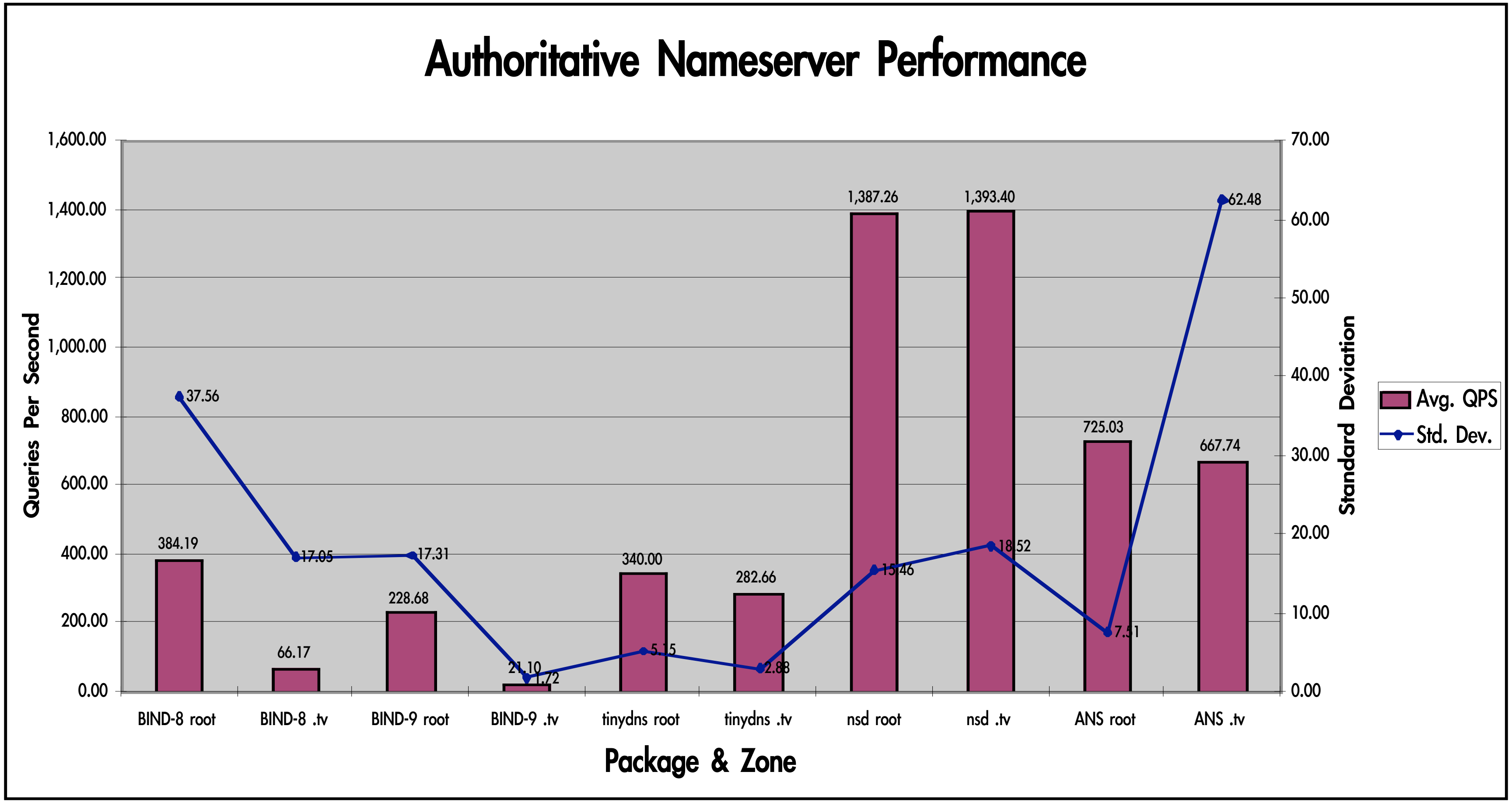
Software:

Performance



Software:

Performance



Conclusions

- BIND is the de-facto Internet standard nameserver, and reference implementation for many advanced DNS features
  - Try to use BIND-9 instead of BIND-8
- However, monoculture environments are susceptible to attack and catastrophic systemic failure

Conclusions

- Need an alternative authoritative-only nameserver?
  - Open Source?
    - SQL database back-end or general-purpose with SQL back-end?
      - Try PowerDNS
        - » Once they have their documentation fixed!
    - Root or TLD nameserver?
      - Try nsd
  - Commercial?
    - Try Nominum Foundation ANS

Conclusions

- Need an alternative caching/recursive nameserver?
  - Commercial
    - Use Nominum Foundation CNS
  - Unfortunately, no suitable open-source alternative at the moment
    - MaraDNS promising?
    - pdnsd promising?
    - Maybe PowerDNS 2.9.5 or 2.9.6?
      - Once they have their documentation fixed!