Improving reverse DNS lookup performance

---Glue in reverse DNS---

Feb. 24, 2005

APNIC19 DNS operations SIG

Kazunori Fujiwara

Japan Registry Services Co., Ltd.

Topics

- Glueless issue in .JP
- The current reverse DNS situation
- Improving reverse DNS lookup performance

Definition of Terms

- In–bailiwick nameserver
  - FQDN with their domain to nameserver
  - Glue is necessary in delegation
- Out–of–bailiwick nameserver
  - FQDN with outside domain to nameserver

.JP case

Outline of a problem in .JP

- JPRS changed the way of handling glue in June 2004
  - This change is described in RFC2181 Section 6.1
  - All of Out-of-bailiwick glues were deleted.
  - As a result, glueless delegations are increased.
- But this causes a problem
  - Some domains make difficulty of name resolution
    - It includes one of the most famous WEB sites in Japan
- We found a problem in BIND8 cache server behavior about glueless delegation processing

BIND 8 caching nameserver behavior

- In iterative query, BIND 8 starts name server hostname resolution at glueless delegation.
- But if all name servers are glueless and all IP addresses are unknown (not in cache), BIND 8 stops first iterative query and does not answer anything.
- After timeout (5 or 10 sec), stub resolver or application re-tries querying to caching server.
- Before then, glueless nameserver addresses may be in cache.
- As a result, name resolution becomes slower for waiting timeouts. (5-30sec)
- At the second time (and after that) DNS query, the DNS cache works well, therefore the problem has been hided.

A worst case: BIND 8.2 Caching Nameserver behabiour

- BIND 8, up to version 8.2.7
- Name resolution fails when glueless delegations twice continuously.

- A typical example
  - EXAMPLE.JP IN NS NS1.EXAMPLE.COM
    - Glueless　once
  - EXAMPLE.COM IN NS NS1.EXAMPLE.ORG
    - Glueless　twice　------> fails here in BIND 8.2
  - EXAMPLE.ORG IN NS NS.EXAMPLE.ORG
    - Glue: NS.EXAMPLE.ORG IN A 192.168.1.1

Other caching nameservers behavior

- The well-known implementations
  - BIND 9
  - dnscache (djbdns)
  - Windows DNS service (2000/2003 server)
- There are no issues of gluelessness

Our report – at NANOG33 meeting

- We reported a problem found in .JP at NANOG33 meeting
  - **“Using In-bailiwick Nameservers”, Masato Minda,** JPRS
  - http://www.nanog.org/mtg-0501/minda.html

- We got a response from Paul Vixie at ISC.

A response from Paul Vixie

- This is a problem caused by older BIND8 and BIND4
- All known workarounds involve providing more glue
  - In-bailiwick glue is an appropriate workaround.
- We regret this inconvenience

The current reverse DNS situation

RFC1912: Common DNS Operational and Configuration Errors

- In section 2.3 “Glue A Records”
- “You shouldn’t have any A records in an in-addr.arpa zone file (unless you’re using RFC 1101-style encoding of subnet masks).”

RIRs/NIRs/LIRs reverse DNS registration

- For example: APNIC Reverse DNS Delegation Form
  - http://ftp.apnic.net/apnic/docs/reverse-dns
  - Nserver object is "List of nameservers for a domain object; a minimum of two is mandatory.Please use fully qualified domain name (FQDN) of the nameserver and not the IP address."

- Reverse DNS registration is limited to FQDN which is outside of "in-addr.arpa" zone.
- As a result, reverse DNS lookup is always glueless.

Why is reverse DNS lookup slow?

- In many cases, reverse DNS lookup is slower than standard DNS lookup.
- The LAME delegation is thought of the most popular cause of this.
- But glueless delegation is certainly the one of the biggest cause of this slow DNS lookup.
  - Most of nameservers in ARPA zone are out-of-bailiwick names and this causes gluelessness.
  - BIND 9 cache server can make reverse DNS lookup much faster than BIND 8 cache server in most cases

A typical example (1)

Lookup 202.11.16.167 (167.16.11.202.in-addr.arpa.)

1. Query “167.16.11.202.in-addr.arpa” PTR to root servers

   root servers (a-m except j) answer 202.in-addr.arpa. Nameservers:

   NS.RIPE.NET. NS1.APNIC.NET. NS3.APNIC.NET. NS4.APNIC.NET.
   DNS1.TELSTRA.NET. TINNIE.ARIN.NET.

   root servers (a,b,c,d,g,l,m) answers with only NS.RIPE.NET glue.

   Almost of nameserver IP addresses are unknown !!

   This glue is special, only BIND8 root servers answer and “NS.RIPE.NET” is outside of “202.in-addr.arpa” zone, it is ignored.

BIND8 starts name server address resolution and stops resolving

-> this causes a client timeout

A typical example (2)

Lookup nameservers IP address from root servers

(In this example, lookup “NS1.APNIC.NET” address)

2. Query “NS1.APNIC.NET” A to root-serves

   root server answers NET nameservers with glue

3. Query “NS1.APNIC.NET” A to .NET serves

   .NET server answers APNIC.NET nameservers with glue

4. Query “NS1.APNIC.NET” A to APNIC.NET servers

   APNIC.NET server answers NS1.APNIC.NET IP address

-> One of “202.in-addr.arpa” servers address is resolved.

A typical example (3)

Return to “167.16.11.202.in-addr.arpa.” PTR lookup.

5. query “167.16.11.202.in-addr.arpa” PTR to NS1.APNIC.NET

NS1.APNIC.NET answers “11.202.in-addr.arpa.” nameservers

a.dns.jp. b.dns.jp.c.dns.jp. d.dns.jp. e.dns.jp f.dns.jp.

All nameserver IP addresses are unknown !!

BIND8 starts name server address resolution and stops resolving -> the cause of the client timeout again

A typical example (4)

Lookup nameservers IP address from root servers
(In this example, lookup “a.dns.jp” address)

6. Query “a.dns.jp” A to root-serves
root server answers .JP nameservers with glue
7. Query “a.dns.jp” A to .JP serves
.JP server answers “a.dns.jp” IP address.

-> One of “11.202.in-addr.arpa” servers address is resolved.

A typical example (5)

Return to “167.16.11.202.in-addr.arpa.” PTR lookup.

8. query “167.16.11.202.in-addr.arpa” PTR to a.dns.jp

a.dns.jp answers

```
16.11.202.in-addr.arpa. IN NS ns01.jprs.co.jp.
16.11.202.in-addr.arpa. IN NS ns02.jprs.co.jp.
```

(some [a-f].dns.jp server answers glue, this is from nameserver software difference.)

All nameserver IP addresses are unknown !!

BIND8 starts name server address resolution and stops resolving -> This causes the client timeout again and again

A typical example (6)

Lookup nameservers IP address: ns01.jprs.co.jp

(In this example, lookup "ns01.jprs.co.jp" address)

But .JP nameservers have been cached at (4). Then

9. Query "ns01.jprs.co..jp" to .JP servers

.JP server answers JPRS nameservers with glue

in this case, ns01.jprs.co.jp address is added as glue.

We should query "ns01.jprs.co.jp" A to ns01.jprs.co.jp, one of authoritative servers of "jprs.co.jp".

-> One of "16.11.202.in-addr.arpa" servers address is resolved.

A typical example (7)

Return to “167.16.11.202.in-addr.arpa.” PTR lookup.

10. query “167.16.11.202.in-addr.arpa” PTR to ns01.jprs.co.jp and ns01.jprs.co.jp answers:

```
167.16.11.202.in-addr.arpa. IN PTR  jprs.jp.
```

- As a result, this reverse DNS lookup requires 10 authoritative server queries.
- And in BIND 8 case, 3 client timeouts occur.
  - “dig” case, default timeout is 5 sec, 3*5sec = 15 sec
- CIDR delegation (especially by using CNAME) needs more queries.
- Real cache server resolves multiple nameservers’ addresses.

Improving reverse DNS lookup performance

Avoid glueless delegation

- My recommendations:
  - Use In-bailiwick nameserver in in-addr.arpa and ip6.arpa.
  - Add glue information to reverse DNS

- For example, 202.11.16.0/24 case
  - 16.11.202.in-addr.arpa domain’s nameserver:
A.NS.16.11.202.in-addr.arpa
B.NS.16.11.202.in-addr.arpa
  - A.NS.16.11.202.in-addr.arpa glue A: 202.11.17.107
  - B.NS.16.11.202.in-addr.arpa glue A: 202.11.17.227

A typical example – all delegations have effective glue

- Reverse DNS lookup finishes three or four queries in most cases and there are no timeouts in BIND 8.
- 202.11.16.167 case, only 4 queries

1. Root server answers APNIC server [202.in-addr.arpa] with glue
2. APNIC server answers JPNIC server[11.202.in-addr.arpa] with glue
3. JPNIC server answers JPRS server[16.11.202.in-addr.arpa] with glue
4. JPRS server answers 167.16.11.202.in-addr.arpa PTR.

This saves cache server cost and decreases resolving time

In-bailiwick nameserver’s Benefits

- Decreasing resolving cost
- Decreasing resolving time

- Using In-bailiwick nameservers removes a dependency of TLDs’ DNS tree.
  - Only depends on root servers and IP number DNS tree
  - It makes easy to troubleshot

Required changes

- Registration system
  - To accept In-bailiwick nameservers
  - To accept glue A/AAAA
- Reverse DNS registration policy
- User's DNS configuration

Another .ARPA case – ENUM

- Using in-bailiwick nameserver on e164.arpa zone is useful similar to in-addr.arpa zone.

Questions?