Large scale database backed DNS operations

Devdas Bhagat <devdas.b@gmail.com>

Requirements

- Midsized domain registrar
  - Slightly under 6 million domains
- Fast updates needed
  - When a domain is added
  - When a domain is removed
  - When any record is changed

Technology choice

- PowerDNS
  - BIND has significant performance issues with a large number of zones
  - BIND with a DB backend has significant performance issues
- PostgreSQL
  - We know this DBMS
- Slony
  - Replication

Why those tools?

- We needed a database backend for ease of management
- PowerDNS is flexible enough that we didn't need custom changes (but if we really need it, we can do so easily).
- PowerDNS is as fast as BIND on a single core, faster with multiple cores
  - http://www.sanog.org/resources/sanog14/sanog14-devdas-dns-scalability.pdf

Things that are WIP

- DNSSEC
  - The version of PowerDNS we used then didn't have DNSSEC support
  - A new version has been deployed, UI for DNSSEC is being worked on.
- Anycast
  - Needs a policy decision from management
  - The primary reason for anycasting would be dealing with DDoS attacks rather than reducing latency.

Stuff that works

- Adding new nodes is easy
  - About a minute of work.
- DNS performance is good
  - We handle about 5000 qps/server across 8 servers
- Record replication latency is measured in seconds, even with large table sizes and replication across the globe
  - Replicated nodes were in Singapore, Hong Kong, Germany and the UK

Stuff that caused problems

- Slony scaling issues
  - Slony defaults to a full mesh system
  - This shows scaling problems around 15 replica nodes
    - Solvable by cascaded replication
- Application level DDoS attacks
  - 500000 qps per server, which is about 10x what each server can do
  - Servers ran out of bandwidth, rather than CPU

?