Building a CDN for Distributing FreeBSD

Allan Jude
allan.jude@scaleengine.com
ScaleEngine Inc.

The current state of Distribution

The FreeBSD Project currently uses a series of donated FTP mirror sites around the globe that provide users with access to the source code, base distributions, ISO images, packages and port distfiles they need to use FreeBSD.

Currently, mirror selection is done manually, users are encouraged to use ftpX.YY.freebsd.org (where X is an integer and YY is a country code).

pkg_add defaults to ftp.freebsd.org which is just a DNS RR of servers at ISC (US West) and TDK (Denmark, Europe)

Looking forward

A number of new trends and demands will strain this existing infrastructure, so now is the time to consider what a new system might look like.

- Expanded selection of distribution formats
  - Existing CD, DVD and boot only ISOs
  - Existing memstick images, plus possibly new USB based distributions (run the system directly off the USB, rather than an installer or livecd)
  - Embedded images for systems such as the raspberry pi (ala pfSense)
  - Virtual machine images (VirtualBox, VMWare, bhyve) (ala PCBSD)

What else do we want?

- Official PKGNG repos
- Packages (PACKAGESITE pkg_add -r)
- Port distfiles
- HTTP Distribution
- GeoIP Aware DNS GSLB
  - Route users to nearby servers (automatic mirror selection)
  - Always return an RR of at least 3 servers
  - Check servers health, avoid directing users to a mirror that is down
  - Check mirror freshness, if a mirror falls behind, stop sending users to it

What we do not want

- Must be backed by multiple vendors and sponsors
  - Avoid vendor lock-in with Amazon or anyone else
  - Rather than using a single commercial CDN, a system that aggregates capacity from donated mirrors, donations from CDNs and possibly paid CDNs
  - Ensure that the CDN cannot go away due to a problem or dispute with a single entity

Building it

A subdomain of freebsd.org delegated to a pool of GSLB DNS servers, such as gslb.freebsd.org

GSLB DNS servers will run gdnsd with a config file programmatically generated from an input list of available CDN nodes following a simple set of ‘business logic’ rules. (this is going to need some additional fleshing out). The idea is to avoid having to require someone to constantly administer the DNS configuration.

Make it Anycast

If possible, the GSLB DNS servers can be Anycast enabled to improve performance and GeoIP accuracy.

In the event of ambiguity or lack of GeoIP data, each anycast server heavily biases traffic to mirrors near its own location, since Anycast will find the topographically nearest DNS server

EDNS-Client-Subnet

EDNS-Client-Subnet is an IETF draft that extends DNS such that recursive DNS servers include the /24 of the requestor in the query to the authoritative server, allowing vastly increased geoip accuracy.

The ultimate goal would also be to get our GSLB DNS servers on the afasterinternet.com whitelist (Google Public DNS and OpenDNS) and any other EDNS-client-subnet whitelists.

Working with existing infrastructure

In the short term the GSLB could also be used to route traffic between the existing FTP mirrors

At this point all that is required is developing the business logic and configuring a number of geographic zones in gdnsd and map the various edge servers to those zones

What do you want?

- What types/formats of distribution might you require in the future?
- What features would you like to see in any new distribution system?
- What pitfalls do you see that I have not?

Other Questions?