ADAPT: A Flexible Solution for Managing the DNS

Jim Reid, Anton Holleman
ORIGIN TIS-INS

Jim.Reid@nl.origin-it.com
Anton.Holleman@nl.origin-it.com

Introduction

- overview of Philips network
- new DNS architecture
  - — rationale
  - — components
    - platforms & tools
- the ADAPT tool
- deploying the new backbone
- on-going problems
- statistics

The Philips Network

- **it's *really* big**
  - **150,000 hosts (+/-50,000)**
  - **around 1500 LANs**
  - **37 Class B networks**
- **no corporate DNS standard**
- **incredible customer diversity**
- **massive global and local DNS roll-outs**

The Philips DNS

- company policy is split DNS
  - — security, access control
  - — billing
- some split DNS on Intranet too
- nameserver zoo
  - — every flavour of UNIX & BIND
  - — NT, Netware, VMS, mainframe
  - — unknown number of DNS tools

The Old DNS Architecture

- it just grew...
- needless diversity
  - — versions of BIND & OS
  - — CM procedures
- resource problems
- root servers on the wrong nets

New DNS Service Design Criteria

- adaptibilty
  - — must meet customer's needs first
  - — cannot impose a solution
- efficiency
  - — fast lookups
  - — low impact on WAN
- scalability
- low cost of ownership
- robustness & simplicity

Review of Existing DNS Service

- **cannot rely on delegated nameservers**
  - **— 24×7 support?**
  - **— local sysadmin procedures vary too much**
- **obvious conclusion:**
  - **— a centrally managed backbone**
    - **identical platforms**
    - **all servers hold everything**

Choosing the DNS Platform

- BSD/OS
  - — obvious choice
  - — BIND development platform
  - — used by big ISPs, Internet root servers
  - — everything just works
  - — nameserver-friendly VM system
  - — commercially supported
- sysadmins wanted Solaris...

Choosing the DNS Software

- **BIND4 or BIND8?**
  - — **BIND4:**
    - known quantity
    - straightforward administration
    - dead end for DNS development
  - — **BIND8:**
    - leap into the unknown?
    - hard to administer
    - poor documentation
    - lots of control hooks
- **Chose BIND8**

Choosing the DNS Hardware

- **Pentium PCs**

  — then the only hardware for BSD/OS...

  — using commodity hardware had attractions

  - **cheap hardware**
  - **easy availabilty of spare parts**
  - **useful for servers in remote locations**

Configuring the DNS Servers

- use systems as DNS toasters
  - — do nothing but run a name server
- main concern for DNS is RAM usage
  - — keep `named` resident
  - — don't want `named` to page fault
- CPU is rarely a bottleneck
- confirmed by the old servers

Memory Requirements

- **simple calculation:**

  — **a DNS RR needs ˜100 bytes**

  — **each IP address has 3 RRs:**

  - **A, PTR and MX**

  — **other stuff lost in the noise**

  — **150,000 × 300 => 45-50 Mb of DNS data**

  — **turned out to be uncomfortably low...**

Locating the Nameservers

- installed in serverfarms
  - — secure computer rooms
    - monitoring, UPS, aircon, etc
  - — good network connectivity
    - backbone routers
    - internet links
    - WWW proxies, SMTP mail relays
  - — serverfarms all over the world
    - Europe
    - North & South America
    - Far East & Australia

Nameserver Installation

- serverfarm locations are fixed
  - — unlikely to move or renumber
  - — should mean no further DNS upheavals
- nameservers are installed in pairs
  - — added redundancy and resilience
  - — allow for server downtime
    - upgrades, field service

DNS Architecture

- root servers and resolver servers
  - logical and functional distinction
  - root server can be a resolver server
  - and *vice versa*
- no caching-only nameservers
  - bad TTLs on important RRs?
  - security concerns - cache pollution
  - propagation of zone updates

Root and Resolver Servers

- root server:
  - — listed in glue records
  - — don’t (need to) recurse
    - already hold everything
- resolver server:
  - — lookups by server farm systems
  - — safety net for customer’s resolvers
  - — can handle recursive queries
    - may see into third-party networks

DNS Server Architecture

- master backbone server
  - — master for all centrally managed & partially delegated zones
  - — slave for everything else
- slave backbone servers
  - — slave for all zones (except root)
- local nameservers
  - — master for delegated zone
  - — slave local data from backbone
  - — may or may not be in glue records

DNS Resolver Architecture

- **resolver servers in serverfarms**
  - **— used by firewalls, WWW proxy servers, SMTP relays, etc**
- **site nameservers**
  - **— should be authoritative for local DNS data**
    - **resilience to WAN failures**
    - **local DNS lookup still works**
  - **— first (best) choice for local systems**
  - **— use nearest resolver server as a fall-back**

DNS Contents Architecture

- **DNS Data**
  - — **SOA, NS records**
  - — **largely static**
- **Mail Routing - MX records**
  - — **rarely changes**
- **Local Site-Specific Data**
  - — **mapping of names and addresses**
    - **PTR, CNAME and A records**
  - — **all the local sysadmin cares about**

Zone Administration

- centrally maintained domains
  - — important top-level domains
    - administered by hand
    - careful CM
- delegated domains
  - — managed by the customer
    - choose their own DNS tool/platform
- partially delegated domains
  - — administered by ADAPT
    - site controls local data
    - backbone controls DNS & MX data

ADAPT

- **Automated DNS Administration Tool**
- **eliminates need for local DNS expertise**
- **painless domain migration**
- **partially delegated domains**

  — local admin supplies names and addresses

  — backbone DNS admin supplies DNS and MX records

- **local admin uses** `/etc/hosts` **file**

  — submitted to backbone with `dnssend`

ADAPT

- **APOP-like authentication with MD5**
  - — no crypto problems
- **file is checked for:**
  - — illegal names
  - — duplicate names and addresses
  - — impossible addresses
  - — local mailhubs and nameservers
- **if OK, transformed into DNS data**
- **email reporting**
  - — prototype WWW interface

ADAPT

- **use `make` to manage zone files**

  — separation between operational and build data

  — reduces human error

- **heavy use of DNS NOTIFY mechanism**

  — fast propagation of updates

Originet Pilot Project

- private IP network for Origin
  - based on NT
  - 15,000 hosts
  - migration from origin.nl and otb.com to origin-it.com
  - removal of old root servers
- DNS architecture fitted perfectly
- good test for ADAPT
- smooth migration
  - worthwhile learning experience

The Reverse Zone Problem

- many sites screw up reverse zones
- build them centrally
    - — clumsy, but effective
    - — no delegations
    - — large zone files
        - 500Kb for 139.130.in-addr.arpa
- needed finer resolution
    - — delegation to trusted sites
    - — ADAPT handles Class C delegations

The Reverse Zone Problem

- split up the reverse zone files
  - — 256 delegations per class B net
  - — added ~10,000 zones (37×256)
- `named` uses another 20 Mb of VM
  - — an extra 10,000 zoneinfo structs
- added an extra 64Mb of RAM to servers
  - — now have 160 Mb of RAM
  - — extra headroom for KEY RRs

Changing the Root Name Servers

- **IP addresses and hardware changes!**
  - — not as bad as first thought
- **5 step procedure:**
  - — use new nameservers in glue records
  - — replace root zone files
  - — disable zone transfers
  - — drop zones from old servers
  - — enable query logging
    - hunt down broken resolvers and servers

On-going Problems

- brain-dead nameserver software
  - — lots of BIND 4.8 nameservers...
  - — no negative caching
  - — aggressive querying of root servers
    - resolvers based on BIND 4.8 don't help
- horrible WINS/DNS interactions
  - — 200,000 queries/day over the WAN for JSPNRMPTGSBSSDIR
  - — very nasty with BIND 4.8

More On-going Problems

- bizarre nameserver configurations

  — think they're on the Internet

  — some even think they're on other private nets!

  — unilateral domain delegation

  — glue record confusion

Yet More On-going Problems

- **broken resolver configurations:**

  — resolve Origin names on Philips servers and *vice versa*...

  — Originet hosts query Philips servers and *vice versa*...

  — misuse of search directive/options

  - **really hurt the root servers**

  — stupid resolvers

  - **try *A.B.C*.philips.com, then**
  - ***A.C*.philips.com ....**
  - ***A*.philips.com, then *A*.**

Even More On-going Problems

- **badly administered delegated domains**

  — local sysadmin doesn't care or know about WAN problems

  — focus on business operations

- **non-standard** `sendmail` **configurations**

- **misconfigured web browsers**

  — don't use proxy WWW servers

  — lookups for:

  - **uls1.four11.com.*XX.YY*.philips.com then**
  - **uls1.four11.com.philips.com, etc, then:**
  - **ULS1.FOUR11.COM.*XX.YY*.philips.com!**

Statistics & Query Analysis

- **DNS on BSD/OS is rock-solid**
  - — almost 100% uptime
- **servers are largely idle**
  - — handle 10-20 queries/second
  - — response time is around 500us
- **only 6 RR types matter:**
  - — A, PTR, CNAME, NS, MX, SOA
    - millions of requests
    - tens of requests for junk like TXT RRs

Future Directions

- support for SRV records
  - — critical for NT5...
- DHCP and DNS integration
  - — big worries about dynamic DNS
    - scaling
    - security
- Secure DNS
  - — embyronic PKI?
- IPv6

Conclusions

- design goals met
  - — costs are low
  - — service is stable
- successful migration to new servers
- no service disruption