Making Internet Services Highly Available

Joe Abley <jabley@isc.org>

These Slides

`http://www.isc.org/misc/netsa2003/ha.pdf`

- You might like to take notes
- These slides will not be a good record of my handwaving, my elaborate whiteboard scribbling or of the useful experience you hear from other people in the room

Agenda

- Availability (what is it?)
- Services, Dependants, Dependencies
- Internet Realities
- Strategies for High Availability
  - Networks
  - Services
- F-Root Name Server

Availability

What is Availability?

- the proportion of the time that something works, expressed as a percentage
  - network link
  - the ability of a service to perform its function

Performance

- Availability is linked to performance
- Needs to be clearly coupled with a test which will indicate whether something is working

Services, Dependants, Dependencies

A Service has...

- a particular job to do
- names or addresses (or both)
- a set of clients
- dependants and dependencies
- availability requirements
- a tendency to become busier
- security requirements

Dependencies

- Service A requires Service B to be available in order to work
  - Service A is a dependant of Service B
  - Service B is a dependency of Service A

Availability Inheritance

- Services inherit availability from their dependencies
  - if Service A depends on Service B, and service B has an availability of 80%, then Service A will never have an availability greater than 80%
  - and so on

Sensible Planning

- There is no point spending time, money and effort making some service reliable if its availability will be crippled by some other unreliable dependency

Maintenance

- We will always need to do maintenance
  - hardware faults
  - software faults
  - operator error
  - growth

Maintenance Policies

- There is no point engineering highly-available services if administrators are allowed to break them accidentally
  - access controls
  - configuration audits
  - maintenance policies (”change control”)

Limits

- Is it possible for a service to have 100% availability?
  - what does that mean?
  - is it worth the cost?
- Is it possible for a network to have 99.999% availability?
  - about 30 minutes downtime per year

Internet Realities

Money

- The more stable and available a service is, the more it costs
  - cost of deployment
    - network costs
    - hardware costs
  - cost of operation

Operations

- The more stable and available a service is, the harder it is to operate
  - more training required for staff
  - operators will make more mistakes
  - mistakes mean downtime

New Features

- The more stable and available a service is, the harder it is to modify
  - new features are harder to roll out
  - needs lots of regression testing
  - migration needs careful planning

User Tolerance

- Most users will tolerate reasonable downtime
  - a couple of hours in the middle of the night, every few weeks, even
  - careful about time zones

Internet Availability

- The Internet itself is not highly available
  - depending on what you mean by “highly available” and “The Internet”
- Hidden Dependency

Plan for Downtime

- If you don't expect unplanned downtime, and plan for it, expect long and unexpected outages
  - panic-driven upgrades
  - wide-scale devastation
  - outages which last days

Strategies for High Availability

Dependencies

- Keep the list of dependencies for all services low
- Avoid single-points of failure
- Make common dependencies highly-available, autonomous and simple
  - dedicated caching resolver servers

User-Visible Components

- Keep the components of services that users interact with directly simple and highly available
- Move as much complexity as possible into dependant components
  - if they fail, users won't notice immediately

Redundant Hardware

- Use redundant hardware, wherever possible
  - if you can transition between hardware components simply, maintenance becomes simple
  - failures can be dealt with on a reasonable schedule, and not in the middle of the night

Reliable vs. Redundant

- You can buy routers, switches and servers which are highly internally redundant
  - hot-swap power supplies
  - hot-swap interface cards, route engines
  - hot-swap CPUs, RAM
  - hot-swap air conditioning units!

Reliable vs. Redundant

- Extremely reliable hardware is extremely expensive
- A redundant array of smaller, less reliable hardware is usually much cheaper
  - plus you get a choice of vendors

Load Balancing

- If you can spread load across multiple components, you can accommodate growth by adding components (”horizontal scaling”)
  - easier than upgrading components (”vertical scaling”)
  - can test and bring into service without touching installed infrastructure

Redundant Software

- Use a couple of different vendors for really critical components
  - protection against systematic software failure
  - e.g. two different caching resolver packages
  - if it doesn't introduce too much operational complexity

Geographic Diversity

- Putting all your servers in one room is fine until
  - the power fails
  - the air conditioning fails
  - earthquake, fire, tsunami, burglary
  - a car drives into the building

Geographic Diversity

- Split-site redundancy
- Diverse network paths
  - diverse fibre paths
  - diverse transmission (e.g. fibre and wireless)
- Widely dispersed sites
  - autonomous points of presence

Simplicity

- Keep the high-availability or load-balancing strategy simple (why?)

Highly-Available Networks

Topological Redundancy

Routing Protocols

- What is Routing?
- What problems do Routing Protocols solve?

Interior Redundancy

- Interior Gateway Protocols (IGPs)
  - OSPF, RIP (eek!), IGRP (eek!), IS-IS, EIGRP (semi-eek!)
- Characteristics of IGPs
- Re-Routing
- Load Balancing

Exterior Redundancy

- The Global Internet Routing System
- Multi-Homing with BGP
- Re-Routing
- Load-Balancing

Gateway-Router Redundancy

Gateway Availability

- Routing protocols allow us to deploy redundant routers to make a reliable network
- Hosts generally have a single default gateway
- The default gateway used by a host is hence a single point of failure for that host

Strategies

- Install multiple default gateways
  - alternate default gateways for similar hosts
  - no session stability
- Install a “virtual” default gateway
  - Hot-Standby Router Protocol (HSRP)
  - Virtual Router Redundancy Protocol (VRRP)

Highly-Available Services

Internet Services

- Distributing some types of service over diverse, redundant components is simple
  - Nameservers
  - Mail Exchangers
- There is no general answer for other types of service
  - SRV records, maybe, one day

General Approach

- Clients use a single name (or address) to direct requests to a server
- Provision multiple servers to answer those requests

Round-Robin DNS

- Install multiple A records for a single service name (don't need service addresses)
- BIND will re-sort the RR set response order, so stupid clients can be happy stupid clients
- Provides approximate load sharing
- Can remove boxes from service for planned maintenance

Round-Robin DNS

- Unexpected failures cause problems
  - if one box out of N fails, 1/N connection attempts will fail
  - changes in the RR set take time to propagate
- Load sharing is inexact
  - greater impact for long-lived sessions, e.g. file transport, streaming media

Layer-4 Switch

- “Layer-7 Switch”
- “Content Switch”
- “Application Switch”
- Arrowpoint (Cisco), Alteon (Nortel), Foundry

Layer-4 Switch

- Service addresses on Switches
- Service health checks
- Load-Sharing strategies

Layer-4 Switch

- Expensive?
- Relatively small number of vendors
- Layer-4 Switch may represent a single point of failure
  - switch upgrades
  - switch expansion

Service Migration

- Use generic service machines
- Service addresses
- Separate “service checker”
- If a service becomes unavailable on one host, automatically start it up on another host
  - move the service address

Service Migration

- Assumes services are simple
- Assumes service code is properly distributed and up-to-date
- Assumes that the service checker works properly, and that the service check is representative of what users see

Service Migration

- Requirement for replicated or network-accessible storage
- single point of failure
- When it breaks, it breaks *everything*
  - this is not fun

Anycast Clusters

- Service Addresses bound to multiple hosts simultaneously
- Run routing protocols on the hosts and routers
  - separate, service-specific area or IGP
- Use CEF or similar to keep session traffic targeted

Common Issues

- Long-lived services vs. short-lived services
  - need to expect that transition events are much less common than client sessions are long
- Stateful services vs. stateless services
  - subsequent transactions might be handled by different servers
  - service protocols need to be atomic

Common Issues

- Anything that requires central storage is hard to distribute
  - unless you separate the storage into an additional component
  - but then that becomes a point of failure for everything
  - availability requirements become huge

Global Service Distribution

- Global as opposed to Local
  - to improve performance
  - to improve reachability
  - to localise attack traffic (maybe)

Nameservers

- Easy
  - good idea to deploy nameservers in different places
- BIND resolvers will choose an appropriate local server
  - Query RTT-monitoring

Mail Exchangers

- Stick MX Hosts in different places
- BIND will rotate members of an equal-cost MX set
- Mail won't necessarily go to the closest MX
  - but MTAs will retry
  - mail isn't interactive anyway

General Services

- Distributing general service types is more difficult, just as with locally distributed clusters
- The added difficulty is that unless we have an internal network which connects all the sites, we are limited in how we can use the routing system to do what we want

Round-Robin DNS

- not served from the closest place
- successive requests might be served by different servers
- failure in one out of N servers will cause 1/N failure rate (see earlier)
- however, we *can* take a box down for planned maintenance

Service Migration

- Service Migration depends on all the services living within the same subnet, so that they can migrate between adjacent boxes
- There is no common subnet here, in the general case
  - even if we made one, the routing would be sub-optimal
- We can't use this

Anycast

- Advertise a covering supernet into the routing system
- Clients see a single server on a single address (the particular server they see depends on where they are)
- Tie the routing advertisement to service performance, and nodes can fall out of service fairly seamlessly

Anycast

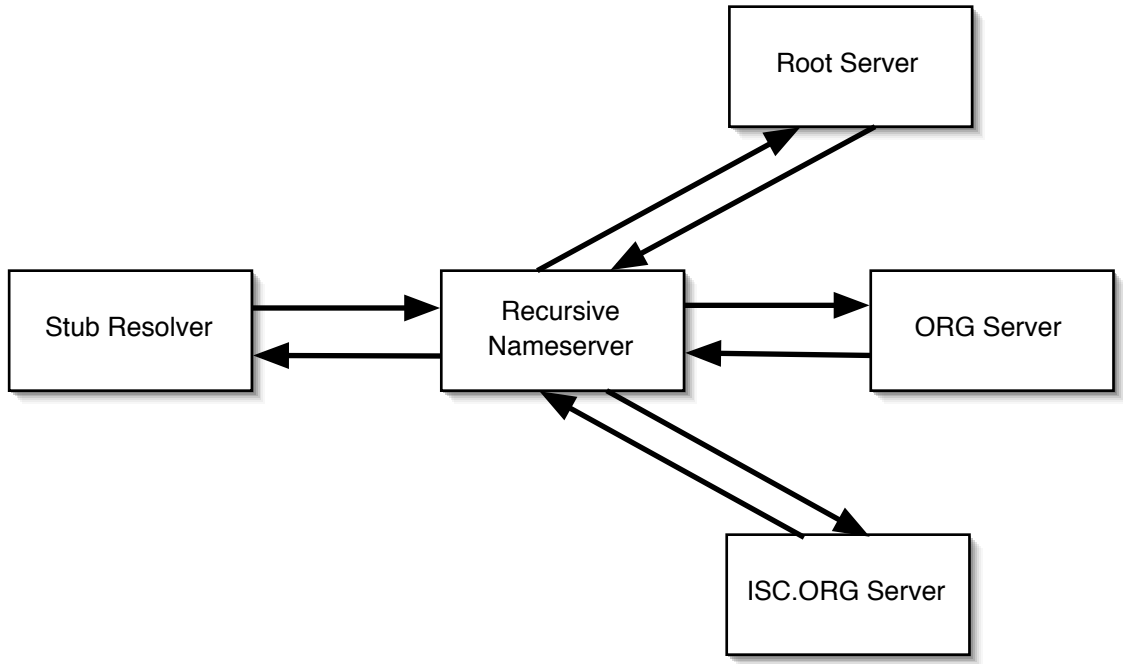
- Can chew up a lot of RIR resources
- Poor address utilisation, probably
- May require an ASN per node, if we are building the nodes to be autonomous
- Requires routers at every node

Case Study

F Root Nameserver

f.root-servers.net (192.5.5.241)

Resolving www.isc.org



Root Servers

- Every recursive nameserver needs to know how to reach a root server
- Root servers are the well-known entry points to the entire distributed DNS database
- There are 13 root server addresses, located in different places, operated by different people
- http://www.root-servers.org/

The Root Servers

A.ROOT-SERVERS.NET	Verisign Global Registry Services	Herndon, VA, US
B.ROOT-SERVERS.NET	Information Sciences Institute	Marina del Rey, CA, US
C.ROOT-SERVERS.NET	Cogent Communications	Herndon, VA, US
D.ROOT-SERVERS.NET	University of Maryland	College Park, MD, US
E.ROOT-SERVERS.NET	NASA Ames Research Centre	Mountain View, CA, US
F.ROOT-SERVERS.NET	Internet Software Consortium	Various Places
G.ROOT-SERVERS.NET	US Department of Defence	Vienna, VA, US
H.ROOT-SERVERS.NET	US Army Research Lab	Aberdeen, MD, US
I.ROOT-SERVERS.NET	Autonomica	Stockholm, SE
J.ROOT-SERVERS.NET	Verisign Global Registry Services	Herndon, VA, US
K.ROOT-SERVERS.NET	RIPE	London, UK
L.ROOT-SERVERS.NET	IANA	Los Angeles, CA, US
M.ROOT-SERVERS.NET	WIDE Project	Tokyo, JP

Challenges on the Root

- There have been a number of attacks on the root servers
- Distributed denial of service attacks can generate a lot of traffic, and make the root servers unreachable for many people
- Prolonged downtime would lead to widespread failure of the DNS

Widespread Failure

- Probability of the entire DNS system failing is low
  - the most important data in the DNS (records which are frequently queried) are cached
- Regional failure is more likely
  - e.g. loss of international connectivity, bulk probe traffic from worms

f.root-servers.net

- Has a single IP address (192.5.5.241)
- Requests sent to 192.5.5.241 are routed to different nameservers, depending on where the request is made from
  - this behaviour is transparent to devices which send requests to F

Hierarchical Anycast

- Some of the F root nameserver nodes provide service for 192.5.5.241 to the entire Internet (global nodes)
  - very large, well-connected, secure and over-engineered nodes
- Others provide service for 192.5.5.241 to a particular region (local nodes)
  - smaller

Hierarchical Anycast

- Architecture described in an ISC Technical Note
  - http://www.isc.org/tn/

Failure Modes

- If a local node fails, queries to 192.5.5.241 are automatically routed to a global node
- If a global node fails, queries are automatically routed to another global node
- Catastrophic failure of all global nodes results in continued service by remote nodes within their catchment areas

Sponsorship

- ISC is a non-profit company
- Equipment, colo, networks for remote nodes are paid for by a sponsor
- All equipment is operated by ISC engineers
- The sponsor covers the ISC's operational costs of running the remote node

Deployment Status

- Two global nodes
  - Palo Alto, CA, US
  - San Francisco, CA, US

Deployment Status

- Auckland, New Zealand
- Hong Kong
- Madrid, Spain
- New York, NY, USA
- Rome, Italy
- San Jose, CA, USA
- Los Angeles, CA, USA

Deployment Targets

- 10 local nodes live by the end of 2003
- 20 more in 2004

For More Information

- Contact ISC
  - Paul Vixie <vixie@isc.org>
  - Joe Abley <jabley@isc.org>
- Contact APNIC
  - Paul Wilson <dg@apnic.net>

`http://f.root-servers.org/`

The End

http://www.isc.org/misc/netsa2003/ha.pdf

Joe Abley <jabley@isc.org>