NetAirt: a DNS-based Redirection System for Apache
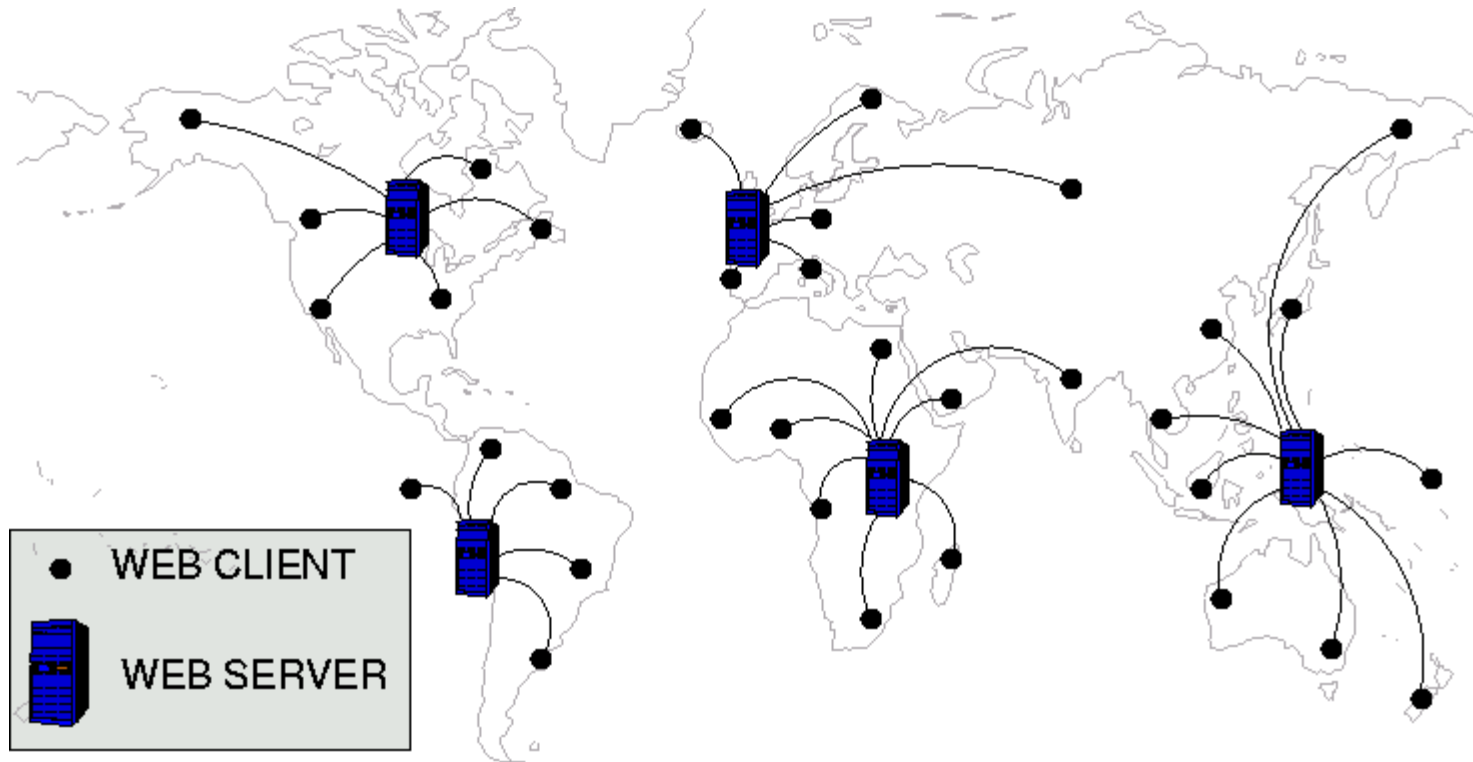
Michal Szymaniak   Guillaume Pierre   Maarten van Steen
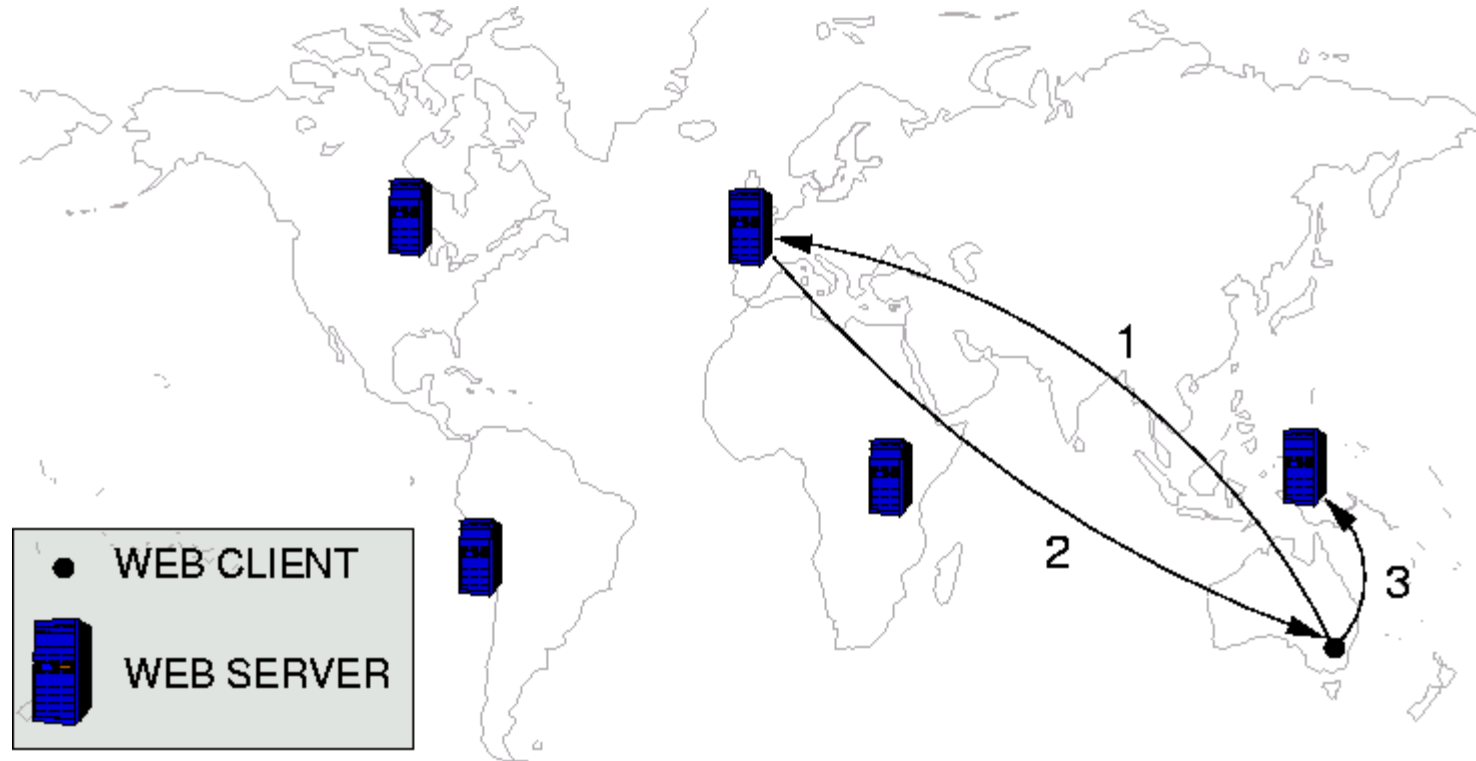
Vrije Universiteit Amsterdam

{michal,gpierre,steen} @cs.vu.nl

Problem

- Popular Web servers are **replicated**
    1. Install many servers (replicas) hosting the same data
    2. Let each replica service its nearby clients
- Data close to the clients ➔ faster access
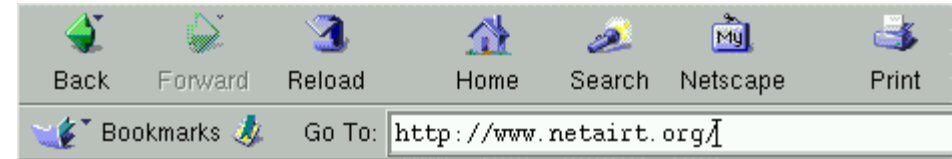- But how do clients find their nearby replicas?

Solution: Redirection

1. Clients first contact the **main** server..

2. ..and obtain an **address** of a nearby replica..

3. ..from which they download the data.

HTTP Redirection Mechanism

1. A client from Sydney enters:
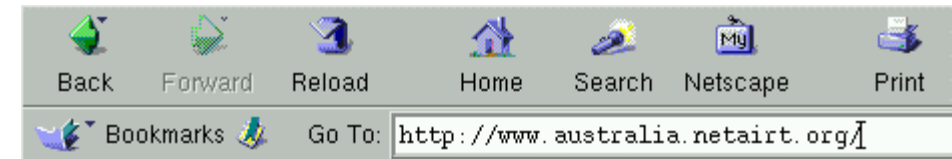
2. **www.netairt.org** responds with:

   GO TO www.australia.netairt.org

3. The client (automatically) connects to:

- Easy to deploy ➜ **simplicity**
- But: clients may become bound to their nearby replica
- What if that replica goes down?

DNS Redirection Mechanism

1. A client asks for the address of **www.netairt.org**

2. The **netairt.org DNS server** responds with:

   `www.netairt.org is 192.168.13.13`

3. The client connects to 192.168.13.13..

   ..which in fact can be **www.australia.netairt.org!**

- Hides the redirection
   ➔ **transparency** ➔ **no bounds**
- Exploits the infrastructure of DNS servers
   ➔ **caching** ➔ **scalability**

Local DNS

?

Root (.) DNS

?

.org DNS

?

.netairt.org DNS

?
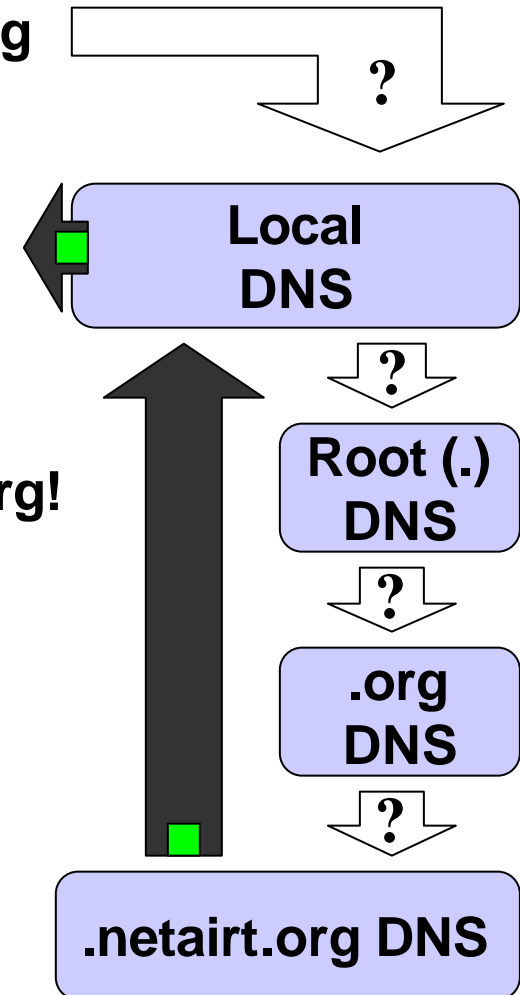
NetAirt: Flexible Redirection

- An Apache module with DNS server functions:
    - DNS packet encoding/decoding
    - UDP datagram handling

- Why Apache module?
    - **Integration**: designed for our Apache-based CDN
        - ➔ entire CDN software inside a single package
        - ➔ other CDN parts know the location of replicas
        - ➔ can still run as a stand-alone redirector
    - **Deployment**: on any Apache-based server

- Flexible:
    - Supports both **HTTP** and **DNS** redirection ➔ all-in-one
    - Separates **<u>Mechanisms</u>** from **<u>Policies</u>** ➔ modular structure

NetAirt Redirection Policies

- Select a replica for each client

- Simple Policies:
  - **Static** - always the same replica - for regular DNS
  - **Round-robin** - A..B..C..A..B..C..A.. - for simple load balancing

- Advanced Policy - **Shortest AS-Path**
  - Each client uses its **topologically-proximal** replica
  - How to measure the distance in the Internet?
  - Hint: **use routing data**

Shortest AS-Path Policy

- The Internet == a graph of Autonomous Systems (AS)

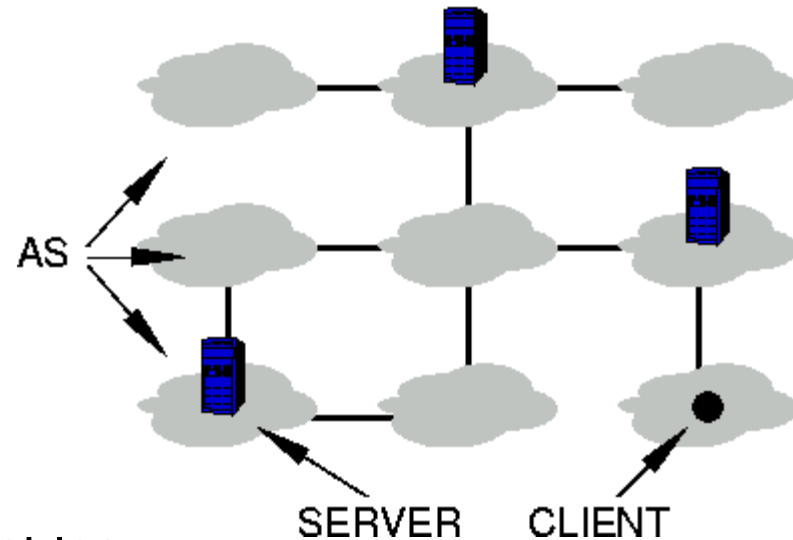- AS - *a set of networks that implement their unique routing policy*

- Only ~**13000** ASes now

- AS-graph is derived from routing tables

- AS-graph describes the Internet **topology**

  → the **number of hops** between two ASes == their **distance**

  → discover the nearby server by applying **Shortest Path** to AS-graph

NetAirt Performance

- DNS Transport Layer (us == microseconds):

Local RTTs	TCP	UDP
NetAirt (DNS)	596 us	392 us
Bind (DNS)	1514 us	821 us
Apache (HTTP)	588 us	n/a

- Shortest AS-Path Policy:
  - Host-to-AS Mapping: avg **1 us**
  - AS-graph Search: max **3.1 ms**, avg **0.64 ms**
    - **Low** compared to the typical name resolution time: **60-200 ms**

Conclusion

- NetAirt - an DNS-based **redirection system** for Apache

- Features:
  - **flexible**
    - **2 mechanisms**: HTTP and DNS
    - **3 policies**: Static, Round-robin, and Shortest AS-Path
  - **scalable and transparent** (because of **DNS** redirection)
  - **easy to deploy**
    - an Apache module
    - runs on a **single** Apache-based Web server

- Soon available from:

    http://www.globule.org/netairt/