Cloud computing and distributed systems

Name services

Zeynep Yücel

Ca' Foscari University of Venice
zeynep.yucel@unive.it
yucelzeynep.github.io

Introduction

- Names for resource identification.
- Enable communication, sharing.
    - Specific names (e.g. URL) for access.
- Consistent naming necessary for sharing.
- Users need recognizable names.
- Descriptive attributes as identification.
    - Name services provide data.
    - Clients can find without names.
    - Entities may change over time.
    - Clients can discover services easily.

Introduction

Names, addresses and other attributes

- Processes need identifiers.
  - Human-readable names: file names, URLs.
  - Identifiers: program-based names.
- Efficient lookup, storage.
- Pure names vs. non-pure names.
  - Pure names: uninterpreted bit patterns, lookup necessary.
- Addresses identify objects, less reliable.

Introduction

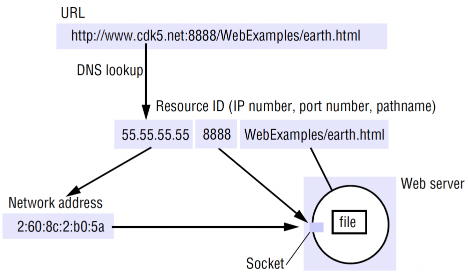
Names, addresses and other attributes

- Name is resolved when it translates to data.
- Binding between name and object.
- Names link to attributes, not implementations.
- Key attribute: object's address.
    - DNS links domain names to host attributes.
        - Browsers use DNS for interpretation.
    - X500 connects names to email attributes.
    - CORBA maps names to references.

Introduction

Names, addresses and other attributes

Composed naming domains used to access a resource from a URL



- An address can be a lookup name.
- Domain names resolve to IP addresses, file systems resolve file names etc.

Introduction

Names and services

- Service-specific names.
  - Used within service context.
  - E.g. Twitter usernames.
- Some names refer to entities beyond the a single scope.
  - E.g. hostnames.
- Middleware uses names for objects.
  - Names must be human-readable.
  - Naming requirements have a global scope.

Introduction

Uniform Resource Identifiers (URIs)

- URIs identify resources on Web or Internet.
  - Aim for coherence in structure.
- Uniform syntax.
- Uniformity simplifies
  - New identifiers.
    - Follow global syntax.
  - Existing identifiers used in other ways.
    - Phone numbers in URIs.
    - Tel URIs for calls.

Introduction

Uniform Resource Locators (URLs)

- URLs provide location access.
  - HTTP protocol.
  - Mailto.
- Downside: outdated links.
  - Leads to dangling linka.

Introduction

Uniform Resource Names (URNs)

- URNs are resource names, not locators.
  - E.g. a URN can identify an email message without info about its storage location.
- URN Format: may start with ‘urn:‘ (or other phrases).
  - ISBN identifies a book.
  - DOI refers to a publication.

Name services and the Domain Name System

- Name service: bindings between names and attributes.
- Such information can be divided in contexts.
- Main function of name service: resolve names.
- Other functions: manage bindings and contexts.

Name services and the Domain Name System

General name service requirements

- Name services were initially very simple.
- Growth and interconection of networks led to challenges.
- Grapevine: An early extensible multi-domain name service.
- Global Name Service goals:
    - Scalability: unlimited names.
    - Long lifetime.
    - High availability.
    - Fault isolation.
    - Tolerance of mistrust.

Name services and the Domain Name System

Name spaces

- Name space: collection of valid names recognized by a service.
- Look up: Resolve names.
  - Rules: Validity distinction.
  - Invalid vs. Valid: '...' vs. 'www.example.com.it' in DNS.
- Structure: Hierarchical names.
- Management: Easier organization.
  - Context: enables relative resolution.

Name services and the Domain Name System

Name spaces

- Context Resolution: Context-based interpretation.
  - Example: '*etc*' in */etc/passwd* is resolved relative to /.
  - '*passwd*' is resolved relative to */etc*.
- Potential Growth: Indefinite expansion.
- Management potentially by diverse authorities.

Name services and the Domain Name System

Name spaces

- Domain names: Hierarchical structure.
  - Components: Strings separated by ‘.’.
- Components connot contain '.' and and cannot be empty.
- Prefixes are initial sections.

Name services and the Domain Name System

Aliases

- Alias: Represents same info under another name.
- Simplifies complex names, alternative name for same entity.
- Example: URL shorteners.
- Domain aliases: One name for another.
    - Update: Only main entry.

Name services and the Domain Name System

Naming domains

- Naming domain: name space with single authority.
- Authority: Overall control.
  - Delegation: Assign responsibility.
- Domains: Consist of names.
  - Example: 'net' contains 'example.net'.
- Main responsibility of authority: database management and update.
- Different domains: distinct name servers and authorities.

Name services and the Domain Name System

Combining and customizing name spaces, Merging

- Global name space: Consistent naming independent of who looks up.
  - Customization: Distinct spaces.
  - Merging, heterogeneity, customization.

Name services and the Domain Name System

Combining and customizing name spaces, Merging

- Merging: One can embed one name space into another, e.g. by merging file systems in UNIX and NFS.
  - Merging file systems
  - Name conflicts possible
  - Create 'super root'
  - Append super root to the beginning of the path
  - What if old names are used (without appended part)
  - Maintain backward compatibility by mounting file systems on all computers
  - Inconvenience in translating old names

Name services and the Domain Name System

Heterogeneity, Customization:

- Heterogeneity:
  - Different name spaces
  - Junctions as mount points
  - E.g.: /.../dcs.qmul.ac.uk/principals/Jean.Dollimore has a junction (principals).
- Customization of Name Spaces:
  - Independent name spaces
  - Translation of names required

Name services and the Domain Name System

Name resolution

- Name resolution process
- Iterative or recursive
- Naming context maps
  - Starts at initial context
    - Example: '/etc/passwd' resolution starts at '/'
  - Iterative alias resolution

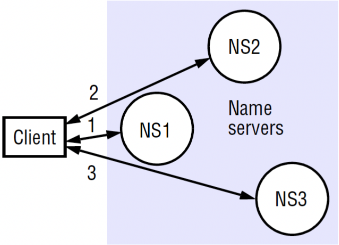
Name services and the Domain Name System

Name servers and navigation

- A large database is not stored centrally
- Avoid bottlenecks and failures; if replicate, also achieve high availability
- How to partition data? By domains
- If data is partitioned, a name server cannot answer all queries without the help of other servers
- Navigation: locating naming data from more than one name server
- Client resolution software

Name services and the Domain Name System

Name servers and navigation



- DNS supports iterative navigation
  - Client sends query to the local name server
  - If it has the name, returns the result immediately
  - Otherwise, suggests another server
  - Further navigation if needed

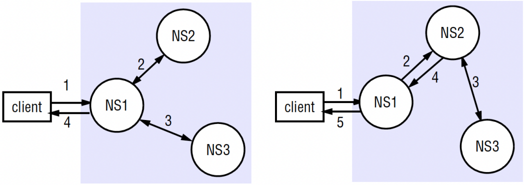
Name services and the Domain Name System

Name servers and navigation

- NFS uses iterative navigation
  - Symbolic links to other servers
- Multicast navigation: client sends requests with multicast
  - Relevant server replies
  - If name is unbound, no response

Name services and the Domain Name System

Name servers and navigation



Non-recursive server-controlled

Recursive server-controlled

- Server-controlled navigation types
- Client chooses a server (e.g. NS1)
  - Non-recursive
    - NS1 asks other servers sequentially, if needed
  - Recursive
    - NS1 leads a chain of requests

Name services and the Domain Name System

Caching

- Caching for name resolution
  - Client checks cache
  - If the cache has the result, return it otherwise send query
  - Servers may return cached data as well
- Improves performance and availability
- Enhances response times
- Bypasses high-level servers
- Convenient, due to infrequent data changes in naming data

Name services and the Domain Name System

The Domain Name System

- Initial naming scheme with single master file had issues
  - Scalability problems
  - No control by local organizations
  - General name service needed
- Replaced by Domain Name System

Name services and the Domain Name System

The Domain Name System

- DNS naming service
- Supports any object naming
  - Primarily names computers (i.e. objects are computers, attributes are IPs)
- Millions of names, hierarchical partitioning of naming database
- Organizations can manage naming data

Name services and the Domain Name System

Domain names

- Hierarchical structure
- Top-level domain on right
- Common top-level domains: com, edu, gov, net, org
- Each country has a domain (us, uk etc)

Name services and the Domain Name System

DNS queries

- DNS main purpose: resolving host names, locating mail host.
- Host Name Resolution: Host names to IPs.
  - Browser.
  - FTP programs.
- Mail Host Location:
  - Email uses DNS for IPs.
  - DNS queries for mail IPs.
- Other
  - Reverse Resolution: IP to domain.
  - Host Information: Stores hardware/software info.

Name services and the Domain Name System

DNS name servers

- DNS scales by partitioning, caching.
- Each server stores local domain data.
- Local queries handled by local servers.
- Other queries forwarded based on recorded server info.
- DNS data divided into zones.
  - Zone: Distinct namespace section managed by a single entity.
- Zone data includes:
  - Attribute data for domain names.
  - Authoritative server info.
  - Authoritative server info for subdomains.
  - Zone-management parameters.

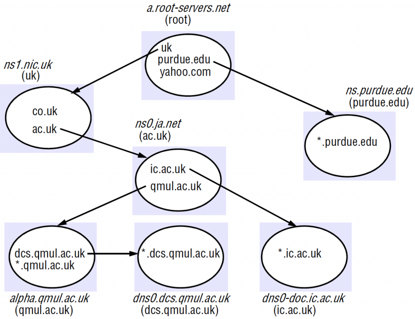
Name services and the Domain Name System

DNS name servers

- Each zone replicated on at least two servers.
- Zone data in a master file.
- Two server types providing authoritative data.
  - Primary: reads from master.
  - Secondary: downloads from primary.
  - Secondary checks for updates periodically.
- Servers that caches data indicates that it is non-authoritative.
  - Non-authoritative caches include time to live.
  - Caches provided only up to that time.

Name services and the Domain Name System

DNS name servers



Name server names are in italics, and the corresponding domains are in parentheses. Arrows denote name server entries

- Root server: first-level, multiple-level.
  - Multiple-level info reduces resolution steps.
- Root holds authoritative entries for top level domains.
- Country domains have own name servers.
- Country servers manage second-level domains.
- University domains manage respective subdomains.

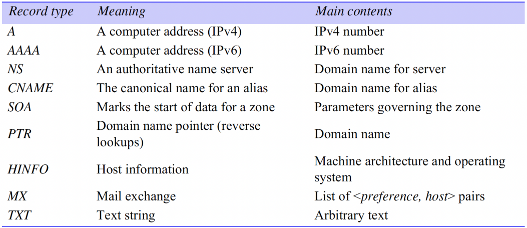
Name services and the Domain Name System

Navigation and query processing

- DNS client: *resolver*.
- Resolver builds queries, formats them into messages and communicates them to name servers.
- Uses simple request-reply protocol.
- If query times out, resolver resends.
- Supports recursive, iterative navigation.
- Allows multiple queries in one request.

Name services and the Domain Name System

Resource records and Load sharing



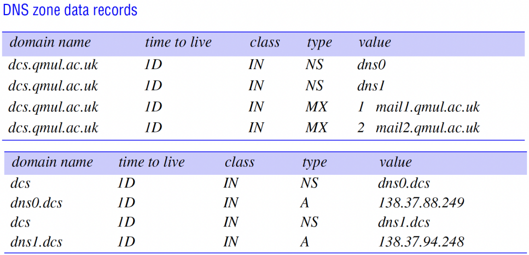
Record type	*Meaning*	*Main contents*
A	A computer address (IPv4)	IPv4 number
AAAA	A computer address (IPv6)	IPv6 number
NS	An authoritative name server	Domain name for server
CNAME	The canonical name for an alias	Domain name for alias
SOA	Marks the start of data for a zone	Parameters governing the zone
PTR	Domain name pointer (reverse lookups)	Domain name
HINFO	Host information	Machine architecture and operating system
MX	Mail exchange	List of <*preference, host*> pairs
TXT	Text string	Arbitrary text

- Zone data: resource records
  - A records: IPv4 addresses
  - AAAA records: IPv6 addresses
  - NS records: name servers
  - SOA records: zone parameters
  - MX records: mail hosts

Name services and the Domain Name System

Resource records and Load sharing

- Other zone record:



DNS zone data records

domain name	time to live	class	type	value
dcs.qmul.ac.uk	1D	IN	NS	dns0
dcs.qmul.ac.uk	1D	IN	NS	dns1
dcs.qmul.ac.uk	1D	IN	MX	1 mail1.qmul.ac.uk
dcs.qmul.ac.uk	1D	IN	MX	2 mail2.qmul.ac.uk

domain name	time to live	class	type	value
dcs	1D	IN	NS	dns0.dcs
dns0.dcs	1D	IN	A	138.37.88.249
dcs	1D	IN	NS	dns1.dcs
dns1.dcs	1D	IN	A	138.37.94.248

- Load sharing: consider a popular website
  - Hosts have same name, but unique IPs
  - Name server keeps all unique IPs
  - Point to hosts in a round-robin fashion

Name services and the Domain Name System

Discussion of the DNS

- Fast lookup response, efficient retrieval
  - Via partitioning, replication, caching
- Primary objects: computers, name servers, hosts
- Hostname-IP mappings have infrequent changes: suitable for caching, replication
- Mail services: per-domain basis
- Limited naming data variety: sufficient for several applications

Directory services

- Name services: $< name, attribute >$ pairs
- Lookup with attributes
- Such service is called a directory service
  - Analogy: yellow pages (business), white pages (phone)
- Directory service: returns matching attributes
  - Example: searches for specific attributes

Directory services

- UDDI provides two way service
- Discovery services for spontaneous networking environments
  - Directory service: structured resource information
  - Discovery service: unpredictable environments
- Attributes are powerful as object designators
- Textual names are simpler for usability

Discussion topic

How does caching help a name service's availability?

Discussion topic

Why do DNS root servers hold entries for two-level names such as yahoo.com and purdue.edu, rather than one-level names such as edu and com?

Discussion topic

Why might a DNS client choose recursive navigation rather than iterative navigation? What is the relevance of the recursive navigation option to concurrency within a name server?