Chapter 9: Name Services

- Introduction
- Name services and the Domain Name System
- Directory and discovery services
- Case study of the Global Name Service
- Case study of the X.500 Directory Service
- Summary

What is Name Service?

- A distinct service that is used by client to obtain attributes such as the addresses of resources or objects when given their names
- Name management is separated from other services
Name and Address

- **Name**
  - A human readable string
- **Identifier**
  - names that are interpreted only by programs.
- **Address**
  - Bits used by machines to locate an object
- **Bind**
  - Association between a name and an address
- **Resolve**
  - Translate from a name to an address
  - **Example**

Attributes

- **Value of a property associated with an object**
- **DNS**
  - IP address
- **X.500**
  - Person’s email address and telephone number
**Uniform Resource Identifiers**

- **URL (Uniform Resource Locator)**
  - Addresses of web resources
  - Dangling problems: a resource may be moved

- **URN (Uniform Resource Name)**
  - Intend to solve the dangling problems
  - E.g. urn:ISBN:0-201-62433-8
  - Urn:net.pku.edu.cn:TR2004-10
  - URN lookup service: mapping from URN to URL

---

**Uniform Resource Identifiers**

- **URC (Uniform Resource Characteristics)**
  - A description of a Web resource (metadata)
Chapter 9: Name Services

- Introduction
- Name services and the Domain Name System
- Directory and discovery services
- Case study of the Global Name Service
- Case study of the X.500 Directory Service
- Summary

Two motivations:

- Unification
  - It is convenient to manage the resource by using the same naming schema

- Integration
  - Separate from the special application system
  - Without a common name service, the administrative domain may use entirely different naming conventions
General name service requirements

• Scalability
  – Arbitrary number of names
  – Arbitrary number of administrative organizations

• Flexibility
  – A long lifetime
  – Accommodate variations on the organization of the set of names

General name service requirements (2)

• High availability
  – Most other systems depend upon it

• Fault isolation
  – Isolate location failures from entire service

• Tolerance of mistrust
  – Not all clients are trusted by all components of the system
Name spaces

- A collection of all valid names recognized by a particular service
- Require a syntactic definition
- E.g. domain name tree of DNS
- E.g. directory tree of a file system
- E.g. flat set of numeric or symbolic identifiers.

Internal structure of a name

- Hierarchic structure, e.g. /etc/passwd
  - Resolve relative to a separate context
  - Potentially infinite
  - Different context managed by different people
- Alias
  - The reason for having aliases is to provide for transparency
  - www.pku.edu.cn( xxx.pku.edu.cn)
Naming domain

• A single administrative authority
  – A name space for which there exists a single authority
  – E.g. pku.edu.cn
  – E.g. cs.pku.edu.cn

• Be stored in a separated server
  – Naming domains are in general stored in different name servers

• Domains, domain’s name domain names

Combining name spaces

• Homogeneous/heterogeneous name spaces
  – DNS is Homogeneous
  – URL is heterogeneous

• Merging
  – E.g. mount file system in Unix and NFS
  – E.g. create a higher-level root context

• Heterogeneity
  – DCE name:
    /.../des.qmw.ac.uk/principals/Jean.Dollimore
    cell principals
    /.../des.qmw.ac.uk/files/pub/reports/TR2000-99
    cell files
Customizing name spaces

- One file with different names
  - E.g. a NFS directory mounted on different machines
- One name refer to different files
  - E.g. install configuration for multi-platform
- One name space per people
  - E.g. Plan 9

Name resolution

- Name space is partitioned in different name servers
- Iterative navigation
  - Client controlling
  - E.g., DNS, NFS
  - A local DNS sever
  Can not reply all the Request with the help Of other servers

A client iteratively contacts name servers NS1–NS3 in order to resolve a name
Server controlled navigation

- Non-recursive/Recursive
  - Recursive type is suitable in environments where administrative domain prohibits client-controlled navigation.

![Diagram of non-recursive and recursive server-controlled navigation]

Caching tech. in name resolution

- Enhance response time
- Eliminate the workload of high-level name servers
- Isolate the failures of high-level name servers
The Domain Name System

• Original Internet Naming scheme
  – A central master files
  – Download to all hosts by FTP

• Domain names [1987]
  – Name space is partitioned both organizationally and according to geography

The DNS name space

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Com</td>
<td>Commercial organizations</td>
</tr>
<tr>
<td>Edu</td>
<td>Universities and other educational institutions</td>
</tr>
<tr>
<td>Gov</td>
<td>US governmental agencies</td>
</tr>
<tr>
<td>Mil</td>
<td>US military organizations</td>
</tr>
<tr>
<td>Net</td>
<td>Major network support centres</td>
</tr>
<tr>
<td>Org</td>
<td>Organizations not mentioned above</td>
</tr>
<tr>
<td>Int</td>
<td>International organizations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domain</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Us</td>
<td>United States</td>
</tr>
<tr>
<td>Uk</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>Cn</td>
<td>China</td>
</tr>
</tbody>
</table>

Even geograph-sounding Domain names are conventional and are completely independent of their physical location.
The DNS name space

- A domain name including the component “.cn” must be looked up within the tree with cn as its root.

DNS queries

- Host name resolution
  - From URL to IP address

- Mail host location
  - Given a domain name, return a list of domain names of hosts that can accept the mail
  - E.g. tom@dcs.rnx.ac.uk
DNS queries …continued

- Reverse resolution
  - From IP to URL
- Host information
  - E.g. the architecture type or operating system of a machine
- Well-known services
  - A list of the services run by a computer
  - Protocol used to obtain them (UDP & TCP)

DNS name servers

- DNS names are divided into Zones
- Zone
  - Include names in the domain, less any sub-domains
  - At least two name servers for the zone
  - Hold name servers for the sub-domains
  - Domain and zones are two different concepts
- Each server hold zero or more Zones
  - Zero zone: the caching name server
  - Time-to-live of secondary
DNS name servers …continued

• Servers that a name server holds
  – Lower-level name servers
    • Child name servers
  – high-level name servers
    • One or more root name servers
    • Parent name server

• There are 13 root server in the world.

DNS name resolution

• Iterative navigation / recursive navigation
  – A query involving three component domain names such as www.berkeley.edu can be satisfied using at worst two navigation steps.

• Example

• DNS resource types
Server controlled navigation

DNS performance

- Replication
  - Zone data are replicated on at least two name servers
  - Master server / secondary server
    - Synchronize periodically

- Cache
  - Any server is free to cache data
  - Time-to-live value
### DNS performance (2)

- **Availability & Scalability**
  - Achieved by a combination of replication, cache and partition
- **Acceptable inconsistent naming data**

### Chapter 9: Name Services

- **Introduction**
- **Name services and the Domain Name System**
- **Directory and discovery services**
- **Case study of the Global Name Service**
- **Case study of the X.500 Directory Service**
- **Summary**
Directory services

- A special kind of naming service
  - Searching attributes
- Entries
  - Each entry is concerned with a set of <attribute, value> pairs
- Query
  - Lookup by known attributes
  - Return interested attributes
  - E.g. query one’s telephone No. by his name

Directory services (2)

- Yellow page / white page
  - Directory service / Conventional naming service
- Directory servers and navigation
  - Similar to name service
- Example
  - Active Directory Service
  - X.500
  - LDAP
Directory services (2)

- Different with DNS
  - Convenient to description
  - No relation with the administrative origination
  - Less simplicity

Discovery services

- A special kind of directory service
- Register the services provided in a spontaneous network
- General operations
  - Register / lookup / de-register
- E.g. a registered printer

```
ResourceClass=printer, type=laser, colour=yes, resolution=600dpi, Location=room101, url=http://www.hotelDuLac.com/services/printer57
```
Jini

• A lookup service
  – Look up services, Jini service, Jini client
  – A service registers an object with a set of attributes
  – Clients query the lookup service
  – Clients download service object that matches query

• Leases
  – A limited period of time during which the service can be used

How to locate lookup service in Jini?

• A priori
• No centralized server
• Multicast to a well-known IP multicast address
  – Lookup services listen on the receiving socket
  – Lookup services announce their existence
  – Different lookup services provide the different multicast address

• Example
Chapter 9: Name Services

• Introduction
• Name services and the Domain Name System
• Directory and discovery services
• Case study of the Global Name Service
• Case study of the X.500 Directory Service
• Summary

Introduction to GNS

• Designed by DEC lab [lampson 1986]
• Design objectives
  – Millions of computer names
  – Billions of email addresses for users
  – Long life time: accommodate changes
Introduction to GNS

• Hierarchical structure
• Name space – organizational structure
• Two assumption:
  – Update is infrequent
  – Slow dissemination of update is acceptance

Architecture of GNS

• **Directory tree / value tree**
• **Directory identifier (DI)**
  – Unique identifier of a directory
• **Name of an entry**
  – <directory name, value name>
  – E.g. <EC/UK/AC/QMW, Peter.Smith/password>
Architecture of GNS (2)

• Multiple name servers
  – Directory tree is partitioned and stored in many servers

• Replication
  – Each partition is replicated in several servers

• Cache
  – Inconsistency cache data is acceptable

How does GNS accommodate changes?

• Merge two name space by a super-root
  – How to it transparent to client applications?
  – E.g. how to locate /uk/ac/qmw ?

Well-known directories:

#599 = #633/EC
#642 = #633/NORTH AMERICA
Working root DI

- **Client agent**
  - Store the DI of working root by client agent
  - E.g., for <UK/AC/QMW, Peter.Smith>, client agent stores #599 which is the DI of “/”, i.e. EC

- **Resolve name**
  - Working root DI + relative path
    - Uniquely refer to a name in the merged tree
    - E.g. <#599/UK/AC/QMW, Peter.Smith>

Well-known directories

- **A table of well-know directories**
  - Mapping between working root DI to new absolute path

- **Replication**
  - Well-know directories are replicated at each nodes
  - Bottleneck of consistency

- **Examples**
Chapter 9: Name Services

- Introduction
- Name services and the Domain Name System
- Directory and discovery services
- Case study of the Global Name Service
- Case study of the X.500 Directory Service
- Summary

X.500 Architecture

- General purpose directory service
  - Directory information tree (DIT)
    - Partitioned and stored in different servers
    - Organized according to distinguished name
  - Service architecture
    - Directory user agent (DUA)
    - Directory service agent (DSA)
Search in X.500

- **DIB entry**
  - Consist of a name and a set of attributes
- **Attribute definition language**
- **Search**
  - Query by “a base name + a filter expression”
  - Return a list of entries

Light Directory Access Protocol (LDAP)

- **A distributed object naming service based on LDAP**
Chapter 9: Name Services

• Introduction
• Name services and the Domain Name System
• Directory and discovery services
• Case study of the Global Name Service
• Case study of the X.500 Directory Service
• Summary

Summary

• Basics of naming service
  – Map between name and attributes of objects
  – binding, resolve
• Name space
  – Syntactic rules
• Multiple name servers
• Cache & replication
• Cases
  – DNS
  – GNS: accommodating changes
  – X.500: directory service
Composed naming domains used to access a resource from a URL

URL

DNS lookup
Resource ID (IP number, port number, pathname)
55.55.55.55 8888 WebExamples/earth.html

Network address
2:60:8c:2:b0:5a file

Web server
Socket

DNS name servers

a.root-servers.net
(root)

ns1.nic.uk
(uk)

co.uk
ac.uk

ns0.ja.net
(ac.uk)

ic.ac.uk
qmw.ac.uk

alpha.qmw.ac.uk
(qmw.ac.uk)

dns0.dcs.qmw.ac.uk
(dcs.qmw.ac.uk)

dns0-doc.ic.ac.uk
(ic.ac.uk)

.purdue.edu

.purdu.edu
(purdue.edu)

uk
purdue.edu

yahoo.com

.co.uk
.ac.uk

* .dcs.qmw.ac.uk

* .ic.ac.uk

### DNS resource records

<table>
<thead>
<tr>
<th>Record type</th>
<th>Meaning</th>
<th>Main contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A computer address</td>
<td>IP number</td>
</tr>
<tr>
<td>NS</td>
<td>An authoritative name server</td>
<td>Domain name for server</td>
</tr>
<tr>
<td>CNAME</td>
<td>The canonical name for an alias</td>
<td>Domain name for alias</td>
</tr>
<tr>
<td>SOA</td>
<td>Marks the start of data for a zone</td>
<td>Parameters governing the zone</td>
</tr>
<tr>
<td>WKS</td>
<td>A well-known service description</td>
<td>List of service names and protocols</td>
</tr>
<tr>
<td>PTR</td>
<td>Domain name pointer (reverse lookups)</td>
<td>Domain name</td>
</tr>
<tr>
<td>HINFO</td>
<td>Host information</td>
<td>Machine architecture and operating system</td>
</tr>
<tr>
<td>MX</td>
<td>Mail exchange</td>
<td>List of <code>&lt;preference, host</code> pairs</td>
</tr>
<tr>
<td>TXT</td>
<td>Text string</td>
<td>Arbitrary text</td>
</tr>
</tbody>
</table>

### Service discovery in Jini

1. ‘finance’ lookup service?
2. Here I am: .....  
3. Request printing service
4. Use printing service
GNS directory tree and value tree for user Peter.Smith

Merging trees under a new root

Well-known directories:

#599 = #633/EC
#642 = #633/NORTH AMERICA
Restructuring the directory

Well-known directories:

#599 = #633/EC
#642 = #633/NORTH AMERICA

X.500 Service Architecture
**Part of X.500 directory information tree**

- X.500 Service (root)
  - France (country)
  - Great Britain (country)
  - Greece (country)
  - BT Plc (organization)
    - University of Gormenghast (organization)
      - Computing Service (organizationalUnit)
        - Department of Computer Science (organizationalUnit)
          - Engineering Department (organizationalUnit)
        - Departmental Staff (organizationalUnit)
          - Research Students (organizationalUnit)
      - Research Students (organizationalUnit)
    - Engineering Department (organizationalUnit)
    - Computing Service (organizationalUnit)
  - Alice Flintstone (person)
  - Pat King (person)
  - James Healey (person)
  - Janet Papworth (person)

**An X.500 DIB entry**

<table>
<thead>
<tr>
<th>info</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
</tr>
<tr>
<td>Alice Flintstone, Departmental Staff, Department of Computer Science, University of Gormenghast, GB</td>
</tr>
<tr>
<td>commonName</td>
</tr>
<tr>
<td>Alice.L.Flintstone</td>
</tr>
<tr>
<td>Alice.Flintstone</td>
</tr>
<tr>
<td>Alice Flintstone</td>
</tr>
<tr>
<td>A. Flintstone</td>
</tr>
<tr>
<td>surname</td>
</tr>
<tr>
<td>Flintstone</td>
</tr>
<tr>
<td>telephoneNumber</td>
</tr>
<tr>
<td>+44 986 33 4604</td>
</tr>
</tbody>
</table>