Distributed Systems

a.y. 2023/2024
Distributed Systems:

Naming
Names:
- Resource sharing
- Unique resource identification
- Reference of locations

Name resolution

Naming systems (distributed across…)

Efficiency and scalability issues
A name is …

- …string of bit (characters) to refer to an entity

- To operate on an entity…access point

- The name of an access point is called address

- Can the name of the access point of an entity be the name of the entity?

- Location independent names
A true identifiers

- An identifier refers to at most one entity.
- Each entity is referred to by at most one identifier.
- An identifier always refers to the same entity.
Forwarding Pointers using (client stub, server stub) pairs

Process P1
Client stub cs

Process P2
Client stub cs*

Stub cs* refers to same server stub as stub cs.

Process P3
Server stub

Identical client stub

Process P4
Object

Interprocess communication

Local invocation

Identical server stub
Home-Based Approaches

1. Send packet to host at its home

2. Return address of current location

3. Tunnel packet to current location

4. Send successive packets to current location

Host's home location

Client's location

Host's present location
Names to address binding

- Human-friendly names
- A (name,address) table over the network…
- Solving names is related to message routing
Distributed Hash Tables

General Mechanism

Resolve $k = 12$ from node 28

Resolve $k = 26$ from node 1
Resolve a key $k$ to the address of $\text{succ}(k)$

A linear approach:

- $p$
- $\text{succ}(p+1)$
- $\text{prec}(p)$
- Each node maintains a table with m entries
- \( FT_p[i] = \text{succ}(p+2^{i-1}) \)
- \( q = FT_p[i] < k < FT_p[i+1] \)
Distributed Hash Tables
General Mechanism

Resolve $k = 12$
from node 28

Resolve $k = 26$
from node 1
Distributed Hash Tables
General Mechanism

- Resolving key 26 from node 1 and key 12 from node 28
- Entering the ring....
  - Lookup for succ(p+1)

- Leaving the ring...
Keeping the table up-to-date

- $q = \text{pred}(\text{succ}(q+1))$ ... For $FT_q[1]$

- $k = q + 2^{i+1}$ for each entry ...
Hierarchical Approaches

- Hierarchical organization of a location service into domains, each having an associated directory node.
Hierarchical Approaches

- Storing information of an entity having two addresses in different leaf domains.
Hierarchical Approaches

Looking up a location in a hierarchically organized location service.

- Node has no record for E, so that request is forwarded to parent.
- Node knows about E, so request is forwarded to child.
- Domain D
- Look-up request

Diagram showing the hierarchical structure and decision-making process in a location service.
Hierarchical Approaches

(a) An insert request is forwarded to the first node that knows about entity E.
(b) A chain of forwarding pointers to the leaf node is created.
A general naming graph with a single root node.
Figure 5-11. The concept of a symbolic link explained in a naming graph.
Information required to mount a foreign name space in a distributed system:
- The name of an access protocol.
- The name of the server.
- The name of the mounting point in the foreign name space.
Mounting remote name spaces through a specific access protocol.
Name Space Distribution
<table>
<thead>
<tr>
<th>Item</th>
<th>Global</th>
<th>Administrative</th>
<th>Managerial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical scale of network</td>
<td>Worldwide</td>
<td>Organization</td>
<td>Department</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>Few</td>
<td>Many</td>
<td>Vast numbers</td>
</tr>
<tr>
<td>Responsiveness to lookups</td>
<td>Seconds</td>
<td>Milliseconds</td>
<td>Immediate</td>
</tr>
<tr>
<td>Update propagation</td>
<td>Lazy</td>
<td>Immediate</td>
<td>Immediate</td>
</tr>
<tr>
<td>Number of replicas</td>
<td>Many</td>
<td>None or few</td>
<td>None</td>
</tr>
<tr>
<td>Is client-side caching applied?</td>
<td>Yes</td>
<td>Yes</td>
<td>Sometimes</td>
</tr>
</tbody>
</table>

Figure 5-14. A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, an administrative layer, and a managerial layer.
Iterative name resolution
Recursive name resolution
## Implementation of Name Resolution (3)

<table>
<thead>
<tr>
<th>Server for node</th>
<th>Should resolve</th>
<th>Looks up</th>
<th>Passes to child</th>
<th>Receives and caches</th>
<th>Returns to requester</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs</td>
<td>&lt;ftp&gt;</td>
<td>#&lt;ftp&gt;</td>
<td>—</td>
<td>—</td>
<td>#&lt;ftp&gt;</td>
</tr>
<tr>
<td>vu</td>
<td>&lt;cs, ftp&gt;</td>
<td>#&lt;cs&gt;</td>
<td>&lt;ftp&gt;</td>
<td>#&lt;ftp&gt;</td>
<td>#&lt;cs&gt; #&lt;cs, ftp&gt;</td>
</tr>
<tr>
<td>nl</td>
<td>&lt;vu, cs, ftp&gt;</td>
<td>#&lt;vu&gt;</td>
<td>&lt;cs, ftp&gt;</td>
<td>#&lt;cs&gt; #&lt;cs, ftp&gt;</td>
<td>#&lt;vu&gt; #&lt;vu, cs&gt; #&lt;vu, cs, ftp&gt;</td>
</tr>
<tr>
<td>root</td>
<td>&lt;nl, vu, cs, ftp&gt;</td>
<td>#&lt;nl&gt;</td>
<td>&lt;vu, cs, ftp&gt;</td>
<td>#&lt;vu&gt; #&lt;vu, cs&gt; #&lt;vu, cs, ftp&gt;</td>
<td>#&lt;nl&gt; #&lt;nl, vu&gt; #&lt;nl, vu, cs&gt; #&lt;nl, vu, cs, ftp&gt;</td>
</tr>
</tbody>
</table>

- Recursive name resolution of `<nl, vu, cs, ftp>`. Name servers cache intermediate results for subsequent lookups.
Attribute-based naming

- \((\text{attribute}, \text{value})\) list
- Directory services
- RDF…resource descriptor framework
LDAP

- Lightweight directory access protocol
an LDAP directory entry…

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Abbr.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>C</td>
<td>NL</td>
</tr>
<tr>
<td>Locality</td>
<td>L</td>
<td>Amsterdam</td>
</tr>
<tr>
<td>Organization</td>
<td>O</td>
<td>Vrije Universiteit</td>
</tr>
<tr>
<td>OrganizationalUnit</td>
<td>OU</td>
<td>Comp. Sc.</td>
</tr>
<tr>
<td>CommonName</td>
<td>CN</td>
<td>Main server</td>
</tr>
<tr>
<td>Mail_Servers</td>
<td>—</td>
<td>137.37.20.3, 130.37.24.6, 137.37.20.10</td>
</tr>
<tr>
<td>FTP_Server</td>
<td>—</td>
<td>130.37.20.20</td>
</tr>
<tr>
<td>WWW_Server</td>
<td>—</td>
<td>130.37.20.20</td>
</tr>
</tbody>
</table>
Hierarchical Implementations: LDAP

Part of a directory information tree.
Hierarchical Implementations: LDAP (3)

- Two directory entries having Host_Name as RDN.
(a) A general description of a resource.
(b) Its representation as an AVTree.
(a) The resource description of a query.
(b) Its representation as an AVTree.
End of lecture