Directory Services

PRINCIPLES – DNS – NIS – LDAP

What is a directory?

Fundamental properties
- Maps keys to values
- Relatively frequent lookups
- Relatively infrequent updates

Examples
- Phone book
- Office directory
- User database
- List of contacts

What is a directory service

A specialized database
- Attribute-value type information
- More reads than updates
- Consistency problems are sometimes OK
- No transactions or rollback
- Support for distribution and replication
- Clear patterns to searches

Directories in Linux

Service names
- /etc/services
- /etc/rpc

Known ethernet addresses
- /etc/ethers

Automount maps
- /etc/auto.master

User database
- /etc/passwd, /etc/shadow

Group database
- /etc/group

Host names
- /etc/hosts

Network names
- /etc/network

Protocol names
- /etc/protocols

The scalability problem

Example
- 13000 users and 5000 hosts
- Passwords valid for 30 days
- 50% of changes made at 8-10
  → One change every 28.8 seconds
  → Propagation time: 0.00567s

Problems
- Performance issues
- Hosts that are down
- Other propagation failures
- Simultaneous updates

Directories in Linux

User database
- /etc/passwd, /etc/shadow

Group database
- /etc/group

Host names
- /etc/hosts

Network names
- /etc/network

Protocol names
- /etc/protocols

Service names
- /etc/services
- /etc/rpc

Known ethernet addresses
- /etc/ethers

Automount maps
- /etc/auto.master

Standard implementation: local files

Directory services

Components
- A data model
- A protocol for searching
- A protocol for reading
- A protocol for updating
- Methods for replication
- Methods for distribution

Common directory services
- DNS
- X.500 Directory Service
- Network Information Service
- NIS+
- Active Directory (Windows NT)
- NDS (Novell Directory Service)
- LDAP (Lightweight X.500)
### Directory services

**Global directory service**
- Context: entire network or entire internet
- Namespace: uniform
- Distribution: usually
- Examples: DNS, X.500, NIS+, LDAP

**Local directory service**
- Context: intranet or smaller
- Namespace: non-uniform
- Distribution: usually
- Examples: NIS, local files

---

### Directory services in Linux

**Alias: name services**
- /etc/nsswitch.conf selects service
- Several services per directory
- Modular design/implementation

Examples from /etc/nsswitch.conf

```
users files,nis
users nis[notfound=return],files
hosts dns,files
```

---

### Network Information Service

**Domain (NIS domain)**
- Systems administered with NIS
- No connection to DNS domain

**NIS server**
- Server that has information accessible through NIS
- Serves one or more domains

**NIS client**
- Host that uses NIS as a directory service for something

---

### NIS

**Protocol**
- RPC based
- No security
- No updates
- Replication support

**Replication**
- Master/slave servers

**Distribution**
- No distribution support!

<table>
<thead>
<tr>
<th>passwd.byname</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Doe</td>
<td>1002:<em>:</em>:0:<em>:0:</em>:0:<em>:1002:</em>:203</td>
</tr>
<tr>
<td>John</td>
<td>1003:<em>:</em>:0:<em>:0:</em>:0:<em>:1003:</em>:203</td>
</tr>
<tr>
<td>Alice</td>
<td>2031:<em>:</em>:0:<em>:0:</em>:0:<em>:2031:</em>:203</td>
</tr>
<tr>
<td>Edward</td>
<td>2032:<em>:</em>:0:<em>:0:</em>:0:<em>:2032:</em>:204</td>
</tr>
<tr>
<td>Rick</td>
<td>2033:<em>:</em>:0:<em>:0:</em>:0:<em>:2033:</em>:204</td>
</tr>
<tr>
<td>Peter</td>
<td>2034:<em>:</em>:0:<em>:0:</em>:0:<em>:2034:</em>:204</td>
</tr>
<tr>
<td>Lara</td>
<td>2035:<em>:</em>:0:<em>:0:</em>:0:<em>:2035:</em>:204</td>
</tr>
</tbody>
</table>

**Data model**
- Directories known as maps
- Simple key-value mapping
- Values have no structure

---

### NIS

**Master server**
- Maps built from text files
- Maps in /var/yp
- Maps built with `make`
- Maps stored in binary form
- Replication to slaves with `ypush`

**Slave servers**
- Receive data from master
- Load balancing and failover

**Processes/commands**
- `ypserv` Server process
- `ypbind` Client process
- `ypcat` To view maps
- `ypmatch` To search maps
- `ypwhich` Show status
- `yppasswdd` Change password
NIS

- NIS client
  - Knows its NIS domain
  - Binds to a NIS server
- Two options
  - Broadcast
  - Hard coded NIS-server
  - ypbind

NIS

Scalability problems
- Flat namespace
- No distribution

Security problems
- No access control
- Broadcast for binding
- Patched as an afterthought

Solution: NIS+

NIS+

Scalability
- Hierarchical namespace
- Distributed administration

Security
- Authentication of server, client and user
- Access control on per-cell level

LDAP

Protocol
- TCP-based
- Fine-grained access control
- Support for updates
- Flexible search protocol

Data model
- Based on X.500
- Object-oriented
- Objects can be extended freely
- Attribute-based data model
- Hierarchical namespace

Example of user

```
NIS+ table "passed.org dir.example.com":

<table>
<thead>
<tr>
<th>name</th>
<th>password</th>
<th>uid</th>
<th>gid</th>
<th>gecos</th>
<th>home</th>
<th>shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>davby</td>
<td><em>LK</em></td>
<td>1211</td>
<td>1200</td>
<td>David</td>
<td>/home/davby</td>
<td>/bin/sh</td>
</tr>
<tr>
<td>fsmith</td>
<td>3x1231v76T89N</td>
<td>1329</td>
<td>1200</td>
<td>Fran</td>
<td>/home/fsmith</td>
<td>/bin/sh</td>
</tr>
</tbody>
</table>

NIS table passwd.byname (user name as key):

<table>
<thead>
<tr>
<th>name</th>
<th>davby</th>
<th>fsmith</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>LK</em></td>
<td><em>LK</em></td>
<td><em>LK</em></td>
</tr>
<tr>
<td>((crypt)3x1231v76T89N</td>
<td>((crypt)3x1231v76T89N</td>
<td>((crypt)3x1231v76T89N</td>
</tr>
<tr>
<td>uidnumber: 1211</td>
<td>uidnumber: 1329</td>
<td>uidnumber: 1329</td>
</tr>
<tr>
<td>gidnumber: 1200</td>
<td>gidnumber: 1200</td>
<td>gidnumber: 1200</td>
</tr>
<tr>
<td>homedirectory: /home/davby</td>
<td>homedirectory: /home/fsmith</td>
<td>homedirectory: /home/fsmith</td>
</tr>
<tr>
<td>loginshell: /bin/sh</td>
<td>loginshell: /bin/sh</td>
<td>loginshell: /bin/sh</td>
</tr>
</tbody>
</table>
```

Example of user

```
ldap://uid=fsmith,ou=employees,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: fsmith
givenName: Fran
sn: Smith
cn: Fran Smith
telephoneNumber: 510-555-1234
roomNumber: 122G
o: Example Corporation International
mailRoutingAddress: fsmith@example.com
mailHost: mail.example.com
userPassword: {crypt}3x1231v76T89N
```
The future

LDAP is taking over
- NIS is too insecure, doesn’t scale and is inflexible
- NIS+ is hard to implement and doesn’t exist on many OSes
- X.500 is too complex and has a bad reputation
- Other options have similar problems

DNS

DNS: Data model
- Functional: \( \text{NAME} \rightarrow (\text{TYPE} \rightarrow \text{RDATA}) \)
- Relational: \( (\text{NAME}, \text{TYPE}, \text{RDATA}) \)

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>RDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ida.liu.se</td>
<td>A</td>
<td>130.236.177.25</td>
</tr>
<tr>
<td>ida.liu.se</td>
<td>MX</td>
<td>0 ida.liu.se</td>
</tr>
<tr>
<td>ida.liu.se</td>
<td>NS</td>
<td>ns.ida.liu.se</td>
</tr>
<tr>
<td>ida.liu.se</td>
<td>NS</td>
<td>ns1.liu.se</td>
</tr>
<tr>
<td>ida.liu.se</td>
<td>NS</td>
<td>ns2.liu.se</td>
</tr>
<tr>
<td>ida.liu.se</td>
<td>NS</td>
<td>nsauth.isy.liu.se</td>
</tr>
</tbody>
</table>

DNS: TYPE & RDATA
- TYPE
  - SOA – Start of authority
  - NS – Name server
  - MX – Mail exchanger
  - A – Address
  - A6 – IPv6 address
  - AAAA – IPv6 address
  - PTR – Domain name pointer
  - CNAME – Canonical name
  - TXT – Text
  - … and many more
- RDATA
  - Binary data, hard coded format
  - TYPE determines format

DNS: Namespace

Names
- Dot-separated parts
- One part after another

FQDN
- Fully Qualified Domain Name
- Complete name
- Always ends in a dot

Partial name
- Suffix of name implicit
- Does not end in a dot

Namespace
- Global and hierarchical

DNS: Replication

Secondary/slave nameserver
- Indicated by NS RR
- Data transfer with AXFR/IXFR

Example
```
sys1-00:~# host -t ns ida.liu.se
ida.liu.se   NS nsauth.isy.liu.se
ida.liu.se   NS ns1.liu.se
ida.liu.se   NS ns2.liu.se
```

Rule of thumb
- Every zone needs at least two nameservers

Questions
- How does a slave NS know when there is new information?
- How often should a slave NS attempt to update?
- How long is replicated data valid?
**DNS: Distribution**

**Delegation**
- A NS can delegate responsibility for a subtree to another NS
- Only entire subtrees can be delegated

**Zone**
- The part of the namespace that a NS is authoritative for
- Defined by SOA and NS

**Domain**
- A subtree of the namespace

---

**DNS: Delegation**

**Delegating NS**
- NS record for delegated zone
- A record (glue) for NS when needed

**Delegated-to NS**
- SOA record for the zone

**Q: When is glue needed?**

---

**DNS: Delegation**

**Format of SOA**
- MNAME Master NS
- RNAME Responsible (email)
- SERIAL Serial number
- REFRESH Refresh interval
- RETRY Retry interval
- MINIMUM TTL for negative reply

**SERIAL**
- Increase for every update
- Date format common
  - 20040909001

**REFRESH/RETRY**
- How often secondary NS updates the zone

**MINIMUM**
- How long to cache NXDOMAIN

---

**DNS: Cacheing**

**TTL**
- Set per RR

**Negative TTL**
- Set in SOA

**Example**

```
@ 1722 in SOA ns.acme.com. root.acme.com 1 7200 60 86400 3600
```

**Choosing good cache parameters is vital**

---

**DNS: The server**

**Recursive/iterative**
- Does the server offer recursion?
- To which clients is it offered?

**Authoritative/nonauthoritative**
- Authoritative: first-hand information
- Otherwise: cached information

**Review**
- Recursive: the nameserver gives a definite answer, but may ask other nameservers in order to generate it
- Iterative: the nameserver gives a definite answer only for locally known information; otherwise it generates a referral

---

**DNS: The client**

**Client requirements**
- Use a recursive NS (resolver)
- Use partially qualified names

**Partially qualified names**
- Add suffix if there are fewer than n dots in the name (ndots)
- Name server (resolver)
  - Specified in /etc/resolv.conf

**Example**

```
# /etc/resolv.conf
search ida.liu.se
nameserver ns.ida.liu.se
ndots 2
```

---
**DNS: Root Name Server**

- **Handles the root zone**
  - Data generated by ICANN
  - Data distributed by Verisign
  - Distribution from hidden master

- **Why no more than 13?**
  - Thirteen services
    - Some are anycast
    - Over 60 servers
  - Hard-coded in resolver

<table>
<thead>
<tr>
<th>Operator</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: VeriSign</td>
<td>Dulles VA</td>
</tr>
<tr>
<td>B: ISI</td>
<td>Marina Del Rey CA</td>
</tr>
<tr>
<td>C: Cogent Communications</td>
<td>Henderson VA; Los Angeles; New York City; Chicago</td>
</tr>
<tr>
<td>D: University of Maryland</td>
<td>College Park MD</td>
</tr>
<tr>
<td>E: NASA Ames</td>
<td>Mountain View CA</td>
</tr>
<tr>
<td>F: Internet Systems Consortium, Inc.</td>
<td>Ottawa; Palo Alto; San Jose, CA; New York City; San Francisco; Madrid; Hong Kong; Los Angeles; Rome; Auckland; Sao Paolo; Beijing; Seoul; Moscow; Taipei; Dubai; Paris; Singapore; Brisbane; Toronto; Monterrey; Lisbon; Johannesburg; Tel Aviv; Jakarta; Munich;</td>
</tr>
<tr>
<td>G: U.S. DOC NTC</td>
<td>Vienna VA</td>
</tr>
<tr>
<td>H: U.S. Army Research Lab</td>
<td>Aberdeen MD</td>
</tr>
<tr>
<td>I: Autonomica/NORDUnet</td>
<td>Stockholm; Helsinki; Milan; London; Geneva; Amsterdam; Oslo; Bangledesh; Hong Kong; Brussells; Frankfurt</td>
</tr>
<tr>
<td>J: VeriSign Global Registry Services</td>
<td>Dulles VA (2 locations); Mountain View CA; Seattle WA; Amsterdam; Atlanta GA; Los Angeles CA; Miami; Stockholm; London; Tokyo; Seoul; Singapore; Sterling VA (2 locations, standby)</td>
</tr>
<tr>
<td>K: RIPE NCC</td>
<td>London; Amsterdam; Frankfurt; Athens; Doha (Quatar)</td>
</tr>
<tr>
<td>L: ICANN</td>
<td>Los Angeles</td>
</tr>
<tr>
<td>M: WIDE Project</td>
<td>Tokyo; Seoul (KR); Paris (FR)</td>
</tr>
</tbody>
</table>

**DNS: CNAME**

- **Canonical name**
  - Pointer within namespace
  - Johansson: See Johnson

**CNAME Woopsie 1**

```
www.CNAME informatix
A  130.236.177.12
```

**CNAME Woopsie 2**

```
ida.liu.se. CNAME www.ida.liu.se.
www.ida.liu.se. A 130.236.177.12
```

**CNAME Woopsie 3**

```
ida.liu.se. NS ns.ida.liu.se.
ns CNAME vitalstatistix
vitalstatistix A 130.236.177.12
```

**DNS: PTR**

- **Address-to-name mapping**
  - Same RR type for IPv4 och IPv6
  - "A big reverse zone in the sky"
  - Reverse address and add in-addr.arpa.
  - Not same as any other name in DNS!
  - Same lookup, cache etc.
  - CNAME works too

```
15.189.236.130.in-addr.arpa. PTR sysi-05.sysinst.ida.liu.se.
```

**DNS: Delegation in in-addr.arpa.**

- **Delegation**
  - Delegation of entire subtrees
  - Subtrees at each dot
  - In in-addr.arpa a dot after each octet of the address

  **Q:** How to delegate partial subtrees corresponding to small subnets, e.g. 10.17.1.0/26?
  **A:** Use CNAME to create a new zone that can be delegated!
  **A:** Delegate each address as a separate zone

  `$GENERATE 1-63 $ CNAME $,rv4.sysinst.ida.liu.se.`

**DNS: The protocol**

- **TCP or UDP**
  - Normally UDP port 53
  - TCP if the reply is too large

- **DNS packet**
  - Header section
  - Flags etc.
  - Query section
  - Queries to the server
  - Answer section
  - Replies to the queries
  - Authority section
  - Pointer to other NS
  - Additional section
  - Extra data that may be useful (e.g. glue)
DNS: The protocol

Header section: flags
- QR Query or response
- Opcode Type of query
- AA Authoritative Answer
- TC Truncation
- RD Recursion Desired
- RA Recursion Available
- Z
- RCODE Result code

Flags
- Set RD for recursive quer
- If AA is not set, reply is from cache
- If TC it set, the reply is too large for UDP

RCODE
- SERVFAIL Problem with NS
- NXDOMAIN No such name
- REFUSED Refuse to reply

Question section
- Contains questions
- Also included in reply

Answer section
- Contains requested RRs
- Empty in referral replies

Authority section
- Indicates authoritative NS
- Never empty in referrals

Additional section
- RR related to response, but not part of response
- E.g. A for NS in authority section

DNS: Commands
- nslookup
- Look up host
- Look up data in DNS

- dig
- Look up data in DNS
- Full access to protocol

- whois
- Information about who has registered a domain
- Many versions – jwhois is nice

sysi-00:~# dig www.ida.liu.se @a.ns.se
; <<>> DiG 9.2.4rc5 <<< >>>
; global options:  printed
; Got answer:
; >>HEADER<<- opcode: QUERY, status: NOERROR, id: 7059
; flags: qr rd:
; QUESTION SECTION:
;www.ida.liu.se.                        IN      A
; AUTHORITY SECTION:
liu.se.                 86400   IN      NS      ns2.liu.se.
liu.se.                 86400   IN      NS      sunic.sunet.se.
liu.se.                 86400   IN      NS      nsauth.isy.liu.se.
liu.se.                 86400   IN      NS      ns1.liu.se.
; ADDITIONAL SECTION:
s2.liu.se.             86400   IN      A       130.236.6.251
sunic.sunet.se.         86400   IN      A       130.236.6.243
nsauth.isy.liu.se.      86400   IN      A       130.236.48.9

sysi-00:~# dig www.ida.liu.se @nsauth.isy.liu.se
; <<>> DiG 9.2.4rc5 <<< >>>
; global options:  printed
; Got answer:
; >>HEADER<<- opcode: QUERY, status: NOERROR, id: 49836
; flags: qr aa rd:
; QUESTION SECTION:
;www.ida.liu.se.                        IN      A
; ANSWER SECTION:
www.ida.liu.se.         259200  IN      CNAME   informatix.ida.liu.se.
informatix.ida.liu.se.  259200  IN      A       130.236.177.26
; AUTHORITY SECTION:
ida.liu.se.             259200  IN      NS      ns1.liu.se.
ida.liu.se.             259200  IN      NS      ns2.liu.se.
ida.liu.se.             259200  IN      NS      nsauth.isy.liu.se.
ida.liu.se.             259200  IN      NS      ns.ida.liu.se.
; ADDITIONAL SECTION:
ns.ida.liu.se.          259200  IN      A       130.236.177.25
ns1.liu.se.             43200   IN      A       130.236.177.23
ns2.liu.se.             43200   IN      A       130.236.177.21
nsauth.isy.liu.se.      21600   IN      A       130.236.48.9

sysi-00:~# dig www.ibm.com @ns.ida.liu.se
; <<>> DiG 9.2.4rc5 <<< >>>
; global options:  printed
; Got answer:
; >>HEADER<<- opcode: QUERY, status: NOERROR, id: 38042
; flags: qr rd ra:
; QUESTION SECTION:
;www.ibm.com.                   IN      A
; ANSWER SECTION:
www.ibm.com.            1800    IN      A       129.42.16.99
www.ibm.com.            1800    IN      A       129.42.17.99
www.ibm.com.            1800    IN      A       129.42.18.99
; AUTHORITY SECTION:
; ADDITIONAL SECTION:
n.s.austin.ibm.com.     70372   IN      A       198.4.83.35
ns.austin.ibm.com.      21600   IN      A       130.236.48.9
**Directory Service Summary**

**Properties**
- Search-optimized database
- Attribute-based data
- Distributed management for scalability
- Replication for performance and reliability
- Search protocol
- Update protocol

**Common directory services**
- DNS – Host names etc.
- NIS/NIS+ – Replace local files
- LDAP – General directory service