Title: Distributed_Applications (06.05.2014)
Date: Tue May 06 14:31:28 CEST 2014
Duration: 88:06 min
Pages: 27

File service

Definition: A file service [Svobodova 1984] provides (remote) centralized data storage facilities to clients distributed among a network. A server deals with bulk data storage, high performance computation, collecting/managing large amounts of data. Client deals with "attractive" display, quick interaction times, use of caching to speed up response time. Use of cache "hints" to facilitate cache management speed up system when hint is correct. In mechanism to detect wrong hint and seek up-to-date information.

Distinction between stateless and stateful
- Stateless server
- Stateful server

Client-server model

Definition: A client-server model implements a sort of handshaking principle, i.e., a client invokes a server operation, suspends operation (in most of the implementations), and resumes work once the server has fulfilled the requested service.

Terms and definitions
- Concepts for client-server applications
- Processing of service requests
- File service
- Time service
  - Definition: A time service provides a synchronized system-wide time for all nodes in the network.
- Name service
- LDAP - Lightweight Directory Access Protocol
- Failure tolerant services
**Definition:** A name service, sometimes called a directory service, provides (remote) centralized name management facilities to clients distributed among a network. Names refer to objects; examples are files, other servers, services, personal computers, printers, as well as users.

Name servers manage a list of names. Such a directory entry might be stored in a data structure: name / Name of the object as parameterized in a client request. */
address / Address of the object within the network, e.g., host number concatenated with communication port number. */
access information / Additional access information may limit access to the object for particular clients. */
attributes / Additional attributes of the object. */

**Domain Name System (DNS):**

Hierarchical domain-based naming scheme for the Internet. 
Distributed database for implementing this naming scheme.

Mapping of host names and email destinations (e.g., www11.in.tum) to their respective IP addresses.

Top-level organizational domains:
- edu: universities and other educational institutions
- com: commercial organizations
- de: organization in Germany

DNS database is distributed across a logical network of name servers.
Each server stores primarily data for the local domain.

**Example for a Name Service**

Generated by Targemsys

---

Example for a Name Service
LDAP is a protocol supporting the access to and update of directory information. It is an open industry standard. LDAP is used by the IntegraTUM project to provide a university-wide directory service at TUM.

**Definition:** A directory is a list of objects arranged in some order and with descriptive information (meta-data).
- directories have a high volume of read requests
- directories do not support transactions
- different query languages

A directory service is a name service containing object names and meta-data.
- Queries in directories: based on names and meta-data.
- White Pages: object access according to object name.
- Yellow Pages: object access according to object meta-data.

LDAP is a communication protocol supporting access to and update of directory information. It has been developed as a simple alternative to X.500 standard.
- It is based on TCP/IP rather than the ISO/OSI protocol stack.
- Modern web browsers (e.g., Netscape) support LDAP.

LDAP specifies several models:
- Information model: basic data structures
- Naming model: referencing of objects (distinguished names)
- Functional model: communication protocol and operations
- Security model: control for directory access

The LDAP architecture is based on the client-server model and the TCP/IP protocols.

1. **General interaction process**
   - Client initiates a session with the LDAP server (binding).
     - Client specifies a name or an IP address and port (e.g., port 389) of the LDAP server.
     - Client specifies username and password.
   - Client invokes LDAP operations (read, write, search).
   - Client terminates session (unbinding).
Examples

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>commonName, cn</td>
<td>cn</td>
<td>name of entry</td>
<td>John Smith</td>
</tr>
<tr>
<td>surname, sn</td>
<td>sn</td>
<td>surname of a person</td>
<td>Smith</td>
</tr>
<tr>
<td>telephoneNumber</td>
<td>tel</td>
<td>telephone number</td>
<td>089-286 25700</td>
</tr>
<tr>
<td>organizationalUnitName, ou</td>
<td>ou</td>
<td>name of organization</td>
<td>Informatik</td>
</tr>
<tr>
<td>organization, o</td>
<td>o</td>
<td>name of organization</td>
<td>TUM</td>
</tr>
<tr>
<td>owner</td>
<td>ou</td>
<td>distinguished name of entry owner</td>
<td>cn=John Smith, o=TUM, o=DE</td>
</tr>
<tr>
<td>jpegPhoto</td>
<td>bin</td>
<td>JPEG3 Photo</td>
<td>photo of John Smith</td>
</tr>
</tbody>
</table>

Based on these attributes, schemas for entries can be defined. Examples of schemas:

- InetOrgPerson: entry for one person
  - attributes: commonName (cn), surname (sn)
- organization: entry for an organization
  - attribute: organization (o)

Naming model

The LDAP naming model defines how entries are identified and organized. Any distinguished name (DN) of an object consists of a sequence of parts, so-called relative distinguished names (RDN).

The entries in an LDAP directory are hierarchically structured as tree (Directory Information Tree, DIT).

Example of DN: cn=John Smith, o=IBM, o=DE.

DIT can be distributed across several servers. Reference to entries of other LDAP servers via URLs.

LDAP - Lightweight Directory Access Protocol

LDAP is a protocol supporting the access to and update of directory information. It is an open industry standard.

LDAP is used by the [IntegraTUM](#) project to provide a university-wide directory service at TUM.

Search

The search operation allows a client to request that an LDAP server search through some portion of the DIT for information meeting user-specified criteria in order to read and list the result(s).

Examples

- find the postal address for cn=John Smith, o=IBM, o=DE.
- find all entries which are children of ou=Informatik, o=TUM, o=DE.

Search constraints:

- base object: defines the starting point of the search. The base object is a node within the DIT.
- scope: specifies how deep within the DIT to search from the base object, e.g.
  - baseObject: only the base object is examined.
  - singleLevel: only the immediate children of the base object are examined; the base object itself is not examined.
  - wholeSubtree: the base object and all of its descendants are examined.
- filter: search filter on entry attributes; Boolean combination of attribute value assertions

Example: $(&!(cn=schmi)(!(cn=de)))$
The client-server model implements a sort of **handshaking principle**, i.e., a client invokes a server operation, suspends operation (in most of the implementations), and resumes work once the server has fulfilled the requested service.

**Terms and definitions**

**Concepts for client-server applications**

**Processing of service requests**

**File service**

**Time service**

**Definition:** A **time service** provides a synchronized system-wide time for all nodes in the network.

**Name service**

**LDAP - Lightweight Directory Access Protocol**

**Failure tolerant services**

There may exist multiple redundant services; server copies and client copies are grouped together into server and client groups.

**Modular redundancy**

**Primary-standby approach**

At any specific time, there is only one replica acting as master (primary replica); RPC requests are always propagated to the primary replica; at checkpoints the current state is propagated to the secondary replicas.

In case of an error the master is replaced by a backup replica.

**Distinction between hot and cold standby.**
Definition: Birell and Nelson (1982) define an RPC as a synchronous flow of control and data passing scheme achieved through procedure calls between processes running in separate address spaces where the needed communication is via small channels (with respect to bandwidth and duration time).

- **synchronous**: The calling process (client) is blocked until it receives the answer of the called procedure (server); the answer contains the results of the processed request.
- **procedure calls**: the format of an RPC call is defined by the signature of the called procedure.
- **different address spaces**: it is necessary to handle pointers during parameter passing different from local procedure calls.
- **small channel**: reduced bandwidth for communication between involved computers.

Neither the client nor the server assumes that the procedure call is performed over a network.

Control flow for RPC calls:

1. **bind to server**
2. **prepare request**
3. **send request**
4. **unpack reply**
5. **RPC-response**
6. **time**

Differences between RPC and local procedure call:

- Basic RPC characteristics
- RPC and OSI
- RPC vs message exchange