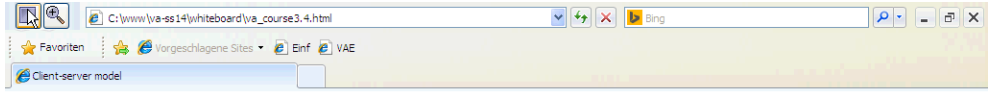
Script **generated by TTT**

Title: Distributed_Applications (06.05.2014)

Date: Tue May 06 14:31:28 CEST 2014

Duration: 88:06 min

Pages: 27



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.

Generated by Targeteam

File service

Definition: A **file service** [Svobodova 1984] provides (remote) centralized data storage facilities to clients distributed among a network.

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.

mechanism to detect wrong hint and seek up-to-date information.

Distinction between stateless and stateful

Generated by Targeteam

Client-server model

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.

Generated by Targeteam

Name service

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	/* This access information may limit access to the object for particular clients. */
attributes	/* Additional attributes of the object. */

Example for a Name Service

Example for a Name Service

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.

Animation Domain Name Service

Animation Domain Name Service

DDomaiNNamS Service

Iterativ — Rekursiv

Example for a Name Service

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.

Animation Domain Name Service

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.

Basics

Definition: A **directory** is a list of objects arranged in some order and with descriptive information (meta-data).

difference between directory and database

directory has 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 / update of directory information.

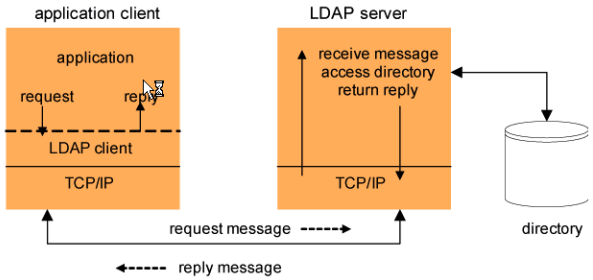
- it has been developed as simple alternative to X.500 standard.
- it is based on TCP/IP rather than the ISO/OSI protocol stack.
- modern web browsers (for example netscape) support LDAP.

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

LDAP architecture

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



LDAP uses strings for data representation.

General interaction process

1. 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 user name and password.

2. Client invokes LDAP operations (read, write, seek).
3. Client terminates session (unbinding).

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.

Basics

LDAP architecture

Information model

Naming model

Functional model

Idif - exchange format

Generated by Targeteam

Examples

Attribute, Alias	*Syntax*	*Description*	*Example*
commonName, cn	cis	name of entry	John Smith
surname, sn	cis	surname of a person	Smith
telephoneNumber	tel	telephone number	089-289 25700
organizationalUnitName, ou	cis	name of organization	Informatik
organization, o	cis	name of organization	TUM
owner	dn	distinguished name of entry owner	cn=John Smith, o=TUM, c=DE
jpegPhoto	bin	JPEG Photo	photo of John Smith

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

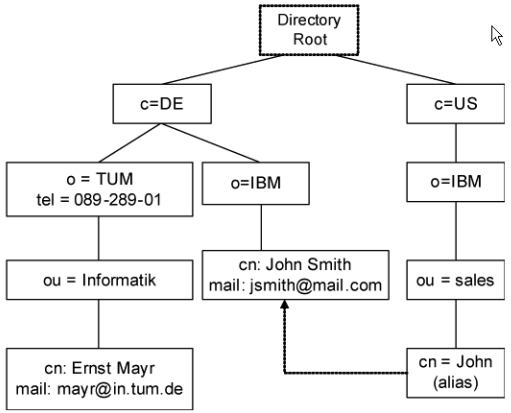
The LDAP naming model defines how entries are identified and organized. Any ***distinguished name*** (DN) of an object consists of a sequence of parts, so-called ***relative distinguished names*** (RDN).

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

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

DIT also supports aliases.

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,c=DE.

find all entries which are children of ou=Informatik,o=TUM,c=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*)(!(c=de)))

Code example

Code example

```
#define SEARCHBASE "o=TUM,c=DE"
LDAP *ld;
char *User = NULL;
char *Passwd = NULL;
char searchfilter[] = "cn=Mayr";
/* open a connection */
if ((ld = ldap_open("ldapserver.in.tum.de", LDAP_PORT)) == NULL) exit(1);
/* authenticate as nobody */
if (ldap_simple_bind_s(ld, User, Passwd) != LDAP_SUCCESS) {
    ldap_perror(ld, "ldap_simple_bind_s");
    exit(1);
}
/* search the database */
if (ldap_search_s(ld, SEARCHBASE, LDAP_SCOPE_SUBTREE, searchfilter, NULL,
0) != LDAP_SUCCESS) {
    ldap_perror(ld, "ldap_search_s");
    exit(1);
}
.....
/* close and free connection resources */
ldap_unbind(ld);
```

Functional model

The functional model defines operations for accessing and modifying directory entries. Among others LDAP supports the following directory operations:

- create a LDAP entry
- delete a LDAP entry
- update a LDAP entry, e.g. modification of the distinguished name (= move in DIT)
- compare LDAP entries
- search for LDAP entries which meet certain criteria

Search

ldif - exchange format

ldif = LDAP Data Interchange Format; it is used to import and export directory information.

```
dn: cn=Informatik
cn: Informatik
objectclass: top
objectclass: groupOfNames
member: cn=Baumgarten,Uwe, mail=baumgaru@in.tum.de
member: cn=Schlichter,Johann, mail=schlicht@in.tum.de
....
dn: cn=Baumgarten,Uwe, mail=baumgaru@in.tum.de
cn: Baumgarten,Uwe
modifytimestamp: 20001213084405Z
mail: baumgaru@informatik.tu-muenchen.de
givenname: Uwe
sn: Baumgarten
objectclass: top
objectclass: person
....
dn: cn=Schlichter,Johann, mail=schlicht@in.tum.de
cn: Schlichter, Johann
modifytimestamp: 20001213084406Z
mail: schlicht@in.tum.de
```

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.

Client-server model

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

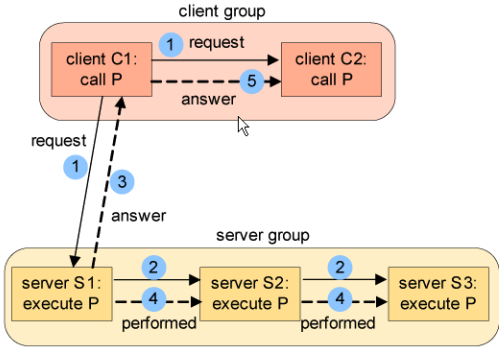
Definition: A **time service** provides a synchronized system-wide time for all nodes in the network.

Name service

LDAP - Lightweight Directory Access Protocol

Failure tolerant services

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.

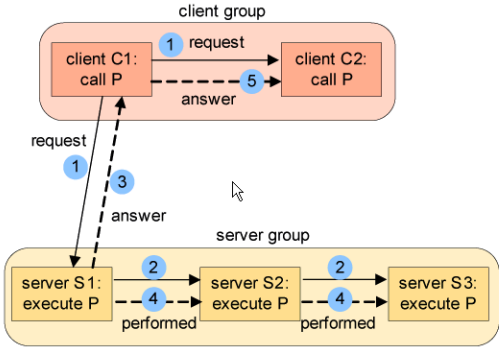
Failure tolerant services

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

Modular redundancy

Primary-standby-approach

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.

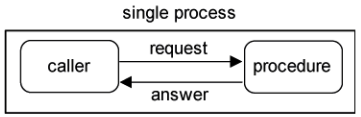
- Prof. J. Schlichter
  - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
  - Boltzmannstr. 3, 85748 Garching

    Email: schlichter@in.tum.de

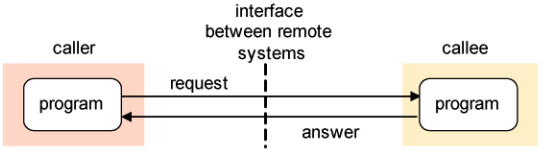
    Tel.: 089-289 18654

    URL: http://www11.in.tum.de/

Local vs. remote procedure call



RPC is an extension of the same type of communication to programs running on different computers; single thread of execution and transfer of data.



Definition: Birrell 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 assume that the procedure call is performed over a network.

Control flow for RPC calls