Scalability in Computer Systems

DNS/BIND as an example

Outline and Goal of Lecture

Outline:

- Scalability:
  terminology, problems and basic approaches
- Names in Distributed Systems:
  purposes of naming, terminology
- Application of scalability approaches on name resolution

Goal:

understand some of the important principles how to build scalable systems

using DNS as an example

Definitions Scalability

Scalability

the ease with which a system or component can be modified to fit the problem area.

http://www.sei.cmu.edu/str/indexes/glossary/

Scalability [in telecommunication and software engineering] indicates the capability of a system to increase performance under an increased load when resources (typically hardware) are added

Wikipedia

Definitions Scalability

Definition(Wang, Xu 98):

- A computer system (HW + SW) is called *scalable* if it can *scale up* (improve its resources) to accommodate ever increasing performance and functionality demand and / or *scale down* (decrease resources) to reduce cost.

Dimensions of Scalability:

- Size (more CPUs)
- Other Resources (Memory)
- Software (Versions, better libs, etc.)
- Heterogeneity
  (different hardware / SW = portability)

Scalability in Computer Systems

A system is described as scalable

if it remains effective when there is a significant increase in the number of resources and the number of users.

Coulouris, Dollimore, Kindberg: Distributed Systems

A system is scalable

if it works well for very large and very small numbers

Another aspect of scalability:

Prepare for change in functionality

- software engineering
- choose sufficiently large logical resources
- provide hooks for extension

Problems for Scalability in Distributed/Parallel Systems

Performance bottlenecks in ...

Failures of ...

Abuse of ...

- computers
- communication

Principles to achieve Scalability (“RPC”)

Partitioning

split systems into parts that can operate independently to a large extent

Replication

provide several copies of components

- that are kept consistent eventually
- that can be used in case of failure of copies

Locality (Caching)

maintain a copy of information

- that is nearer, cheaper/faster to access than the original

Some Challenges

Balance load:

keep load under reasonable threshold

- at each component
- in the communication subsystems

load balancing can be static or dynamic. Will study a detailed example for dynamic load balancing later(Mosix).

Minimize the delay induced by “RPC”.

Prepare for change.

Information Dissemination.

Choose right degree of consistency.

Names, Identifiers, Addresses

Names

- symbolic
- have a meaning for people

Identifiers

- identifies a component (uniquely)
- are used by programs

Addresses

- locates a component
- can change

Name resolution

Name Resolution:

- map symbolic names to objects
- better: to a set of attributes such as identifiers, addresses, other names, security properties

Interfaces:

- Register (Name, attributes, ...)
- Lookup (Name) -> attributes

Related

Compilers

- statically map names to addresses

Dynamic libraries

- dynamically remap addresses

Port Mapper

- map service to port

Name resolution is a form of dynamic mapping of pathnames to attributes.

Observation

Many services, tools, ... provide their own name resolution

- file systems
  path names to I-Nodes
- login
- RPC systems (portmapper)

Purpose of Directory Services

- integration of name services
- generic name service
- world-wide use of names

Today mostly used:

  - email/web
  - computer attributes (IP addresses)
  - people attributes (certificates, ...)

A Bit of History

UUCP/MMDF (cum grano salis):

- ira!gmdzi!oldenburg!heinrich!user (path to destination)
- user@ira!heinrich%gmdzi
  (mixing identifiers and path information)

ARPA-Net:

- a single file: hosts.txt
- maintained at Network Information Center of SRI (Stanford)
- accessed via ftp
- TCP/IP in BSD Unix => chaos
  name collisions, consistency, load

DNS: Paul Mockapetries (84) ...

More Terminology

Name Space

- set of names recognized by a name service

Context

- unit for which a name can be mapped directly

Aliases

- several names for one object

More Terminology

Naming Domain

- Subtree in the hierarchy of DNS contexts

Zone

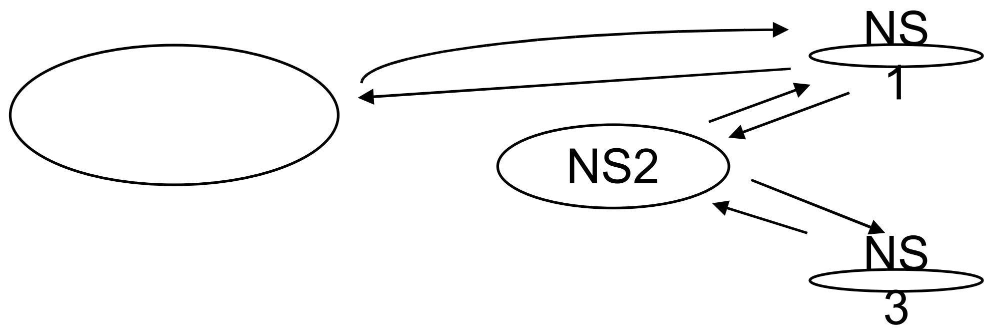
- (aka Zone of authority) Subset of a domain over which an authority has complete control. Subzones (starting at apices of a zone) can be delegated to other authorities.

Navigation

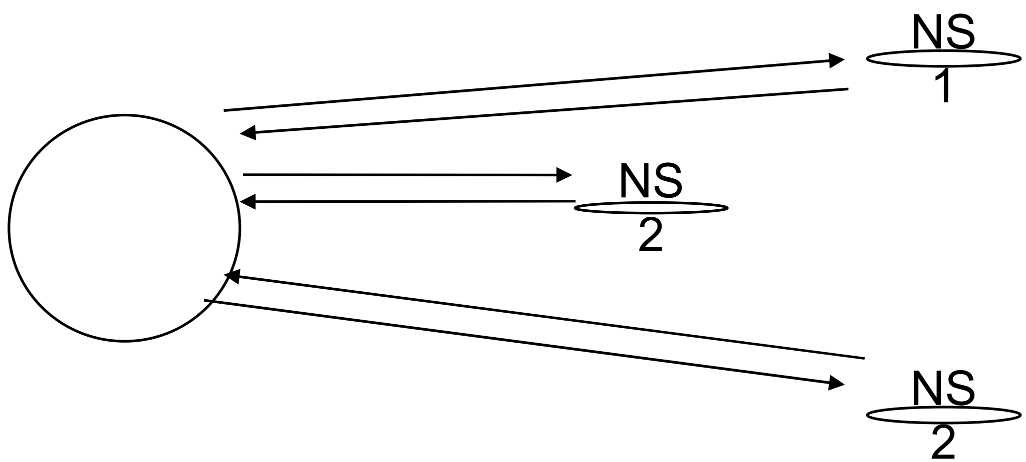
- querying in a set of cooperating name spaces

Basic Implementation Variants

recursive



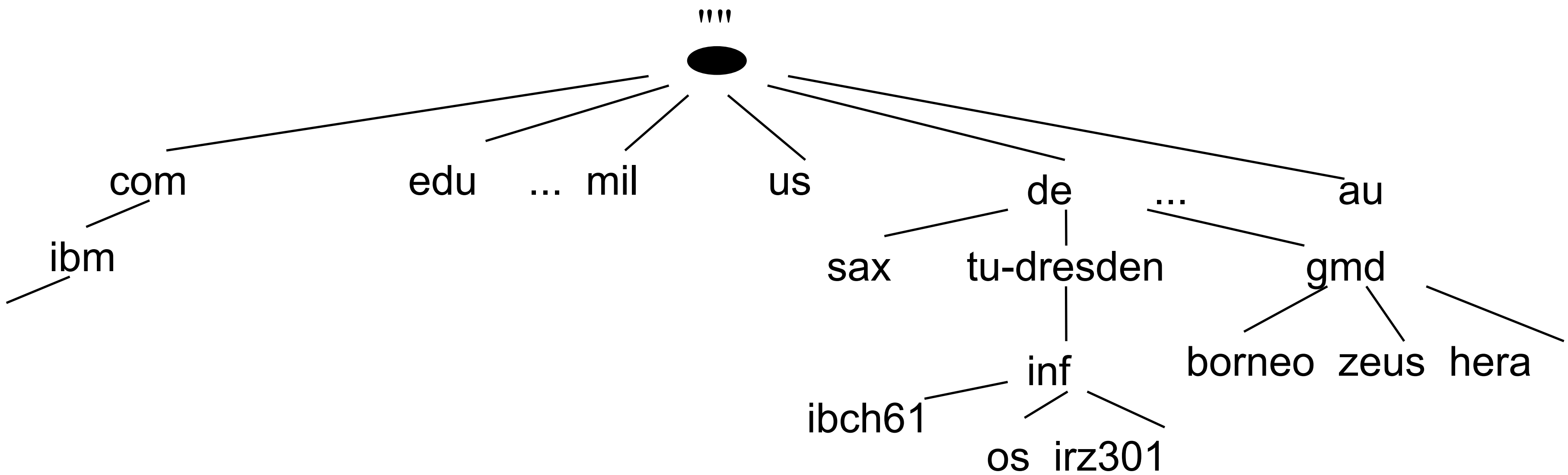
iterative



Requirements / Properties

- arbitrarily large numbers
- arbitrary units of administration
- long living names, the higher in the hierarchy the longer
- high robustness
- restructuring of name spaces
- consistency
- efficiency

DNS Name Space



Examples

inf.tu-dresden.de Domain

os.inf.tu-dresden.de Computer

heidelberg.ibm.com Domain

ftp ftp.inf.tu-dresden.de

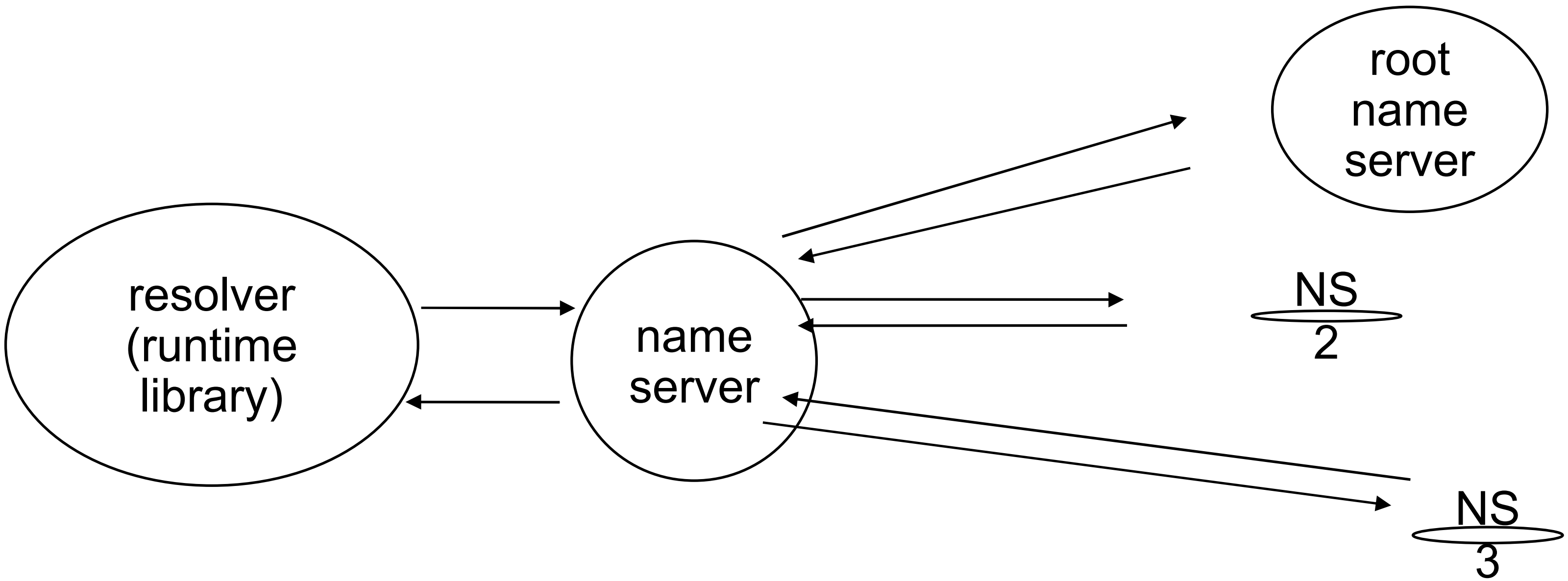
DNS: --> IP address: 141.76.2.3

ftp daemon: IP address, port 21

Properties:

- location independent
- not very deep

Implementation Structure (BIND)



Partitions: Zones

Zones:

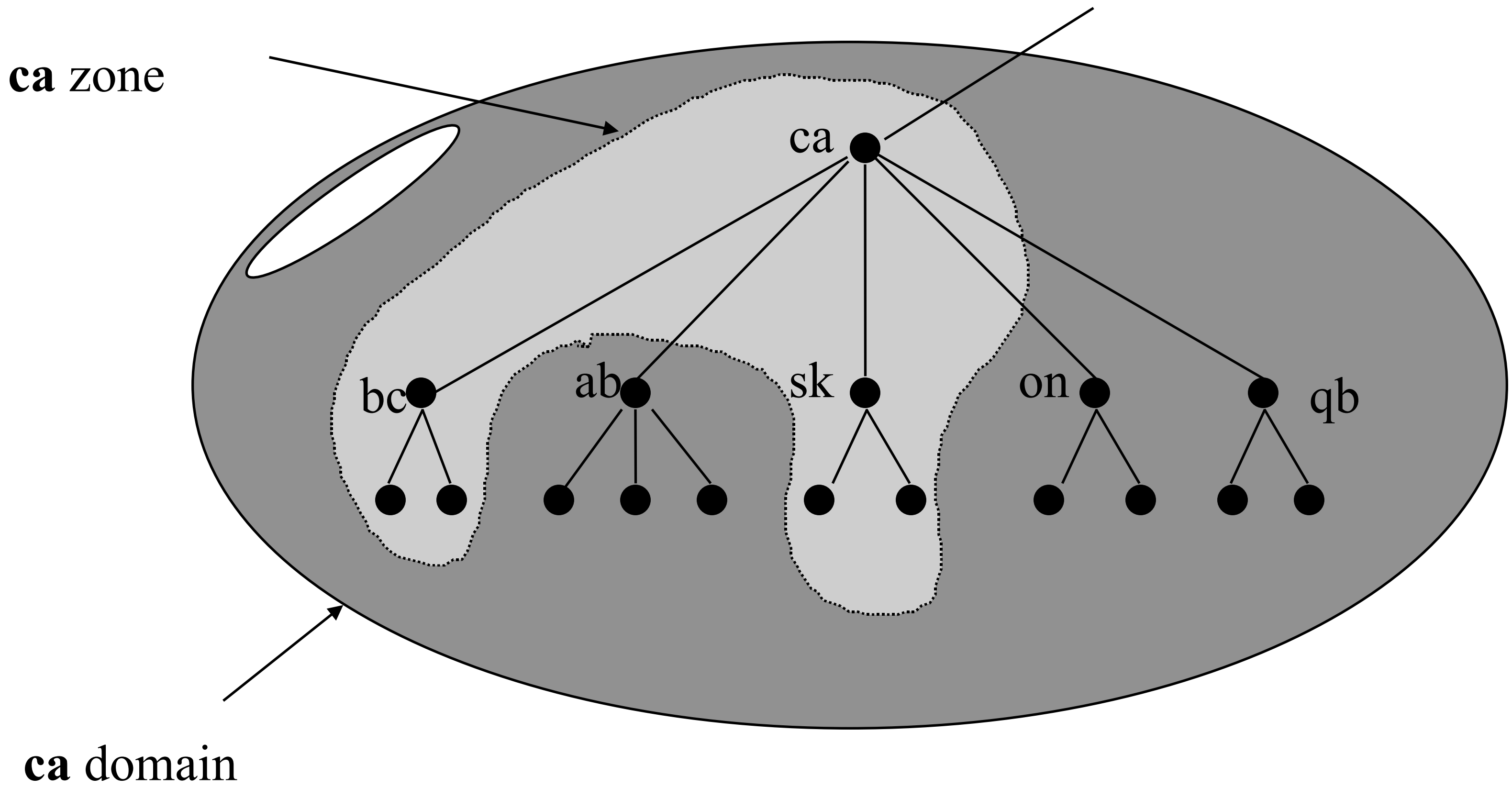
- administrative unit
- resolves all names within a zone recursively
- maps to names and addresses of name servers responsible for sub zones
- maintains management data

Name server:

- process doing the name resolution for one zone

Resource records (RR):

- key interface


ca zone
ca
bc
ab
sk
on
qb
ca domain

Replication

Currently 13 root name servers

each zone has at least

- one primary
- one secondary

name server

Caching

each name server caches resource records

time to live attribute

authoritative versus non-authoritative answers

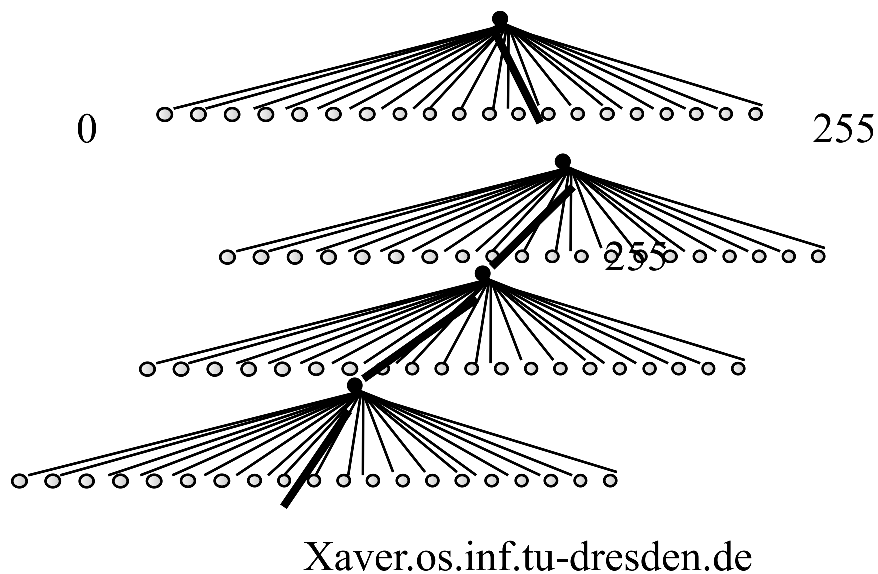
Resource Records

record type	interpretation	content
A	address	IPv4 address
AAAA	address	IPv6 address
NS	name server	DNS name
CNAME	symbolic link	DNS name of canonical name
SOA	start of authority	zone-specific properties
PTR	IP reverse pointer	DNS name
HINFO	host info	text description of host OS

...

Reverse Resolution

Example
IP-Address: 141.76.48.97
=> DNS-Name: 97.48.76.141.in-addr.arpa



Summary: Scalability and DNS

Good points:

- replication and caching work well
- over time, DNS scaled from small numbers to millions

Bad Points:

- IP addresses too small

Literature

Paul Albitz & Cricket Liu

DNS and BIND

O´Reilly & Associates, Inc.