

# *Direccionamiento en Sistemas Distribuidos II*

*Prof. Wilmer Pereira*  
*UCAB / USB / UCV*

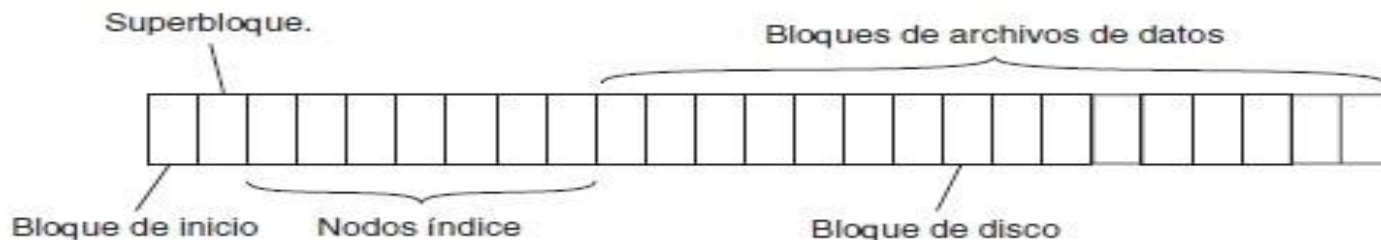
# Nombramiento

Las estrategias de organización más utilizadas son: estructurado en árboles, tablas de *hash* distribuidas (DHT) y nombramiento por atributos

- La estructura en árboles, tipo jerárquico, son las que usa, por ejemplo, el sistema de archivo de linux
- Las tablas de *hash* aplican a cualquier tipo de base de datos donde se pretenda hacer búsqueda eficiente. De hecho existe una propuestas de DNS organizado con DHT
- El nombramiento por atributos tiene su principal representante en sistemas distribuidos con LDAP

# File System de UNIX

- La ruta a cada recurso se arma desde la raíz (vinculo absoluto) o desde cualquier parte del árbol (vinculo relativo) siempre desde lo más general a lo más específico.
- En UNIX cada recurso puede ser:
  - ➔ Archivo de texto o binario,
  - ➔ Directorio
  - ➔ Dispositivo externo: disco, pendrive, lector de DVD
  - ➔ Canal de comunicación externo (*socket*) o interno (*pipe*)
  - ➔ Enlace físico o simbólico
- Todo recurso es un inodo con información sobre:
  - ➔ Espacio que ocupa el recurso en disco,
  - ➔ Fecha de creación
  - ➔ Permisos
  - ➔ Apuntadores a los bloques de memoria



# *Direcccionamiento de recursos*

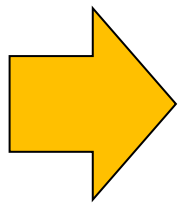
- Sobre el directorio `/mnt` se pueden montar dispositivos externos como discos de alguna máquina de la LAN. Existe el protocolo NFS (*Network File System*) mediante nombramiento estilo URL

PROTOCOLO://DOMINIO/RECURSO

<nfs://www.usb.ve/~wpereira/>

# Implementación Espacio de nombres

## ○ Por capas



Global (a nivel de la WAN)

Administración (a nivel de la LAN)

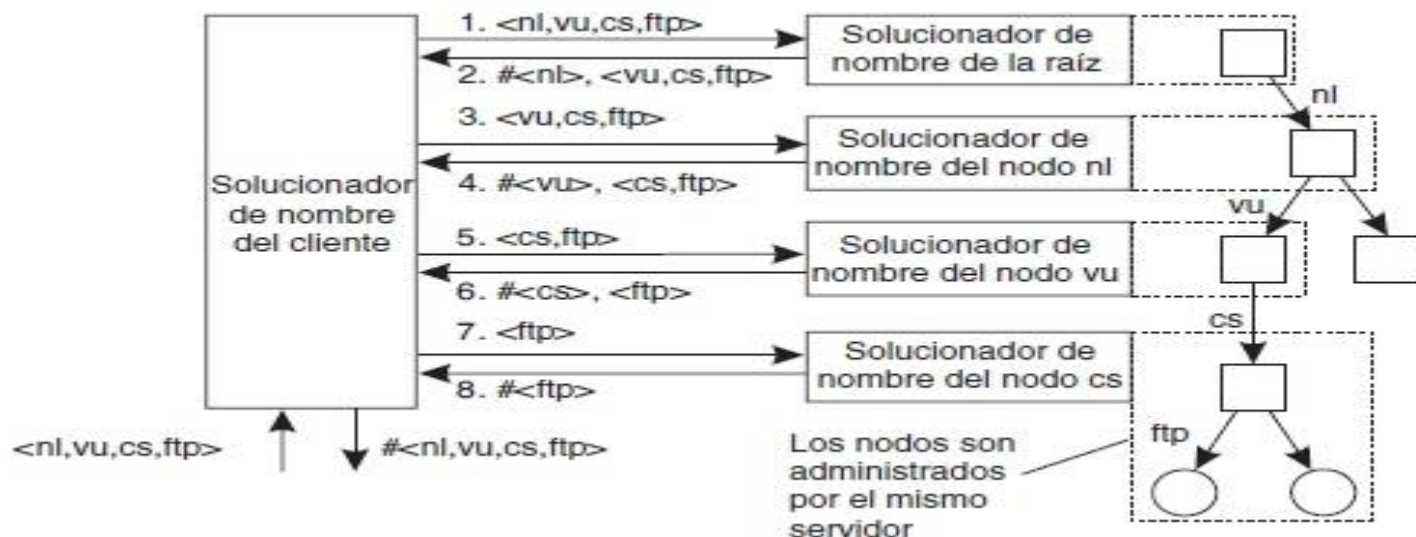
Dirección (dentro de un sistema distribuido localizado)

Elemento	Global	Administración	Dirección
Escala geográfica de una red	A nivel mundial	Organización	Departamento
Número total de nodos	Pocos	Muchos	Cuantiosos números
Respuesta a las búsquedas	Segundos	Milisegundos	Inmediata
Propagación de actualizaciones	Lenta	Inmediata	Inmediata
Número de réplicas	Muchas	Ninguna o pocas	Ninguna
¿Se aplica el cacheo del lado del cliente?	Sí	Sí	Ocasionalmente

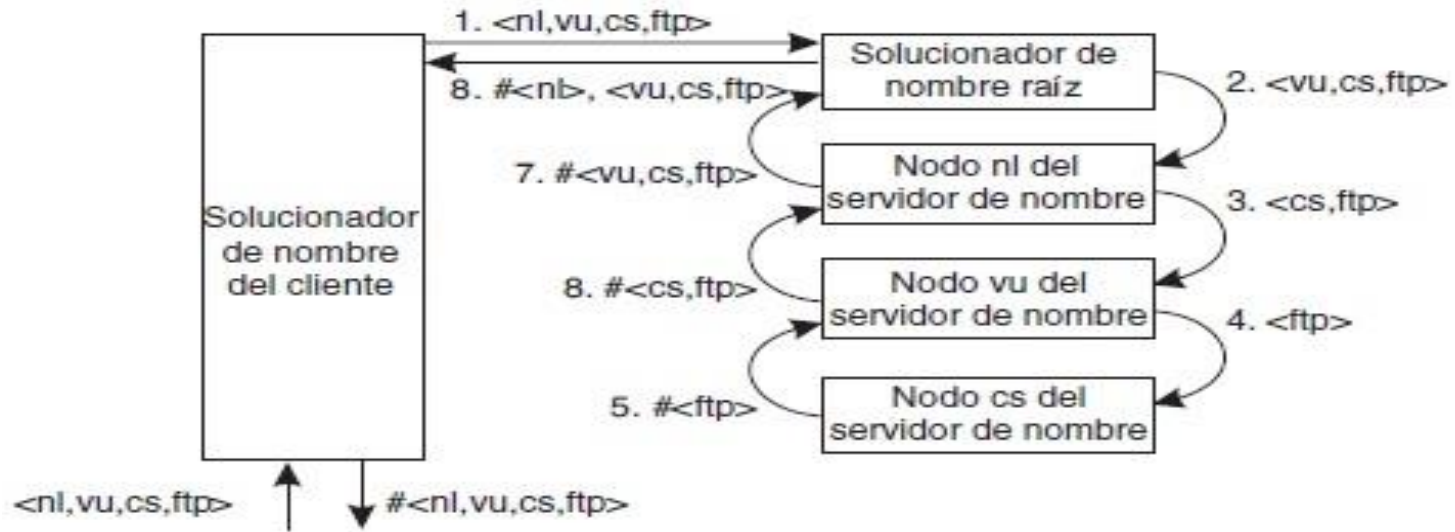
# Resolución de nombres

El objetivo es buscar lo más eficientemente posible el recurso en una base de datos distribuida

- Iterativa: el cliente consulta explícitamente cada nodo para resolver la consulta
- Rekursiva: el cliente delega en los nodos la búsqueda del recurso



# Resolución de Nombres



- En recursiva cada servidor trabaja más ... sin embargo tiene la ventaja de que el cache que instale es más útil para hacer las búsquedas cada vez más eficientes. Además los costos de comunicación son menores
- En iterativa el trabajo de búsqueda es más responsabilidad del cliente por lo que descarga la red ... pero ... es más ineficiente desde el punto de vista de la red ... Sin embargo al servidor raíz le conviene la resolución iterativa ...

# DNS con DHT

DHT mapea eficientemente nombres planos, sin jerarquía  
... pero ...  
se pierden búsquedas asociadas al árbol: hijos de un nodo

- CoDo usa DHT empleando los prefijos de las claves para buscar un nodo
- Se selecciona un prefijo  $b$  de la clave y supongamos que el nodo con clave 3210 mantiene la tabla de enrutamiento de:

$n_0, n_1, n_2, n_{30}, n_{31}, n_{33}, n_{320}, n_{322}$  y  $n_{323}$

- Este nodo es responsable de consultas con prefijo 321. Si recibe consulta de 3123 lo reenvía a  $n_{31}$  ...
- 3210 replica su información en el nodo 321 y a su vez al nodo 32 hasta el nodo 3 ... es decir todo nodo replica a  $i$  prefijos coincidentes. Como no se pueden tener demasiadas replicas se duplica hasta que la latencia sea menos a  $C$ .



# Nombramiento por atributos

El usuario busca por características (atributos) de una entidad más que por un nombre que puede ser plano o estar en una estructura

- Inicialmente, el decidir que atributos son los relevantes puede ser complicado, por ejemplo, videos o música ...
- RDF (*Ressource Definition Famework*), que utiliza la Web semántica, emplea tripletas (sujeto, predicado, objeto) mezclado con ontologías (jerarquía de conceptos con atributos y relaciones) para enriquecer la metadata asociada a las páginas Web

El cielo es azul  $\Leftrightarrow$  (cielo,color,azul)

- A diferencia de los nombres estructurados por jerarquía, se debe acceder a los recursos por varios items o descriptores, dispersos en Internet y almacenados de diferentes maneras

# LDAP

## *Lightweight Directory Access Protocol*

Proviene del standard X500 de OSI donde cada registro son pares (*atributo,valor*) con un tipo asociado pudiendo ser monovaluados o multivaluados

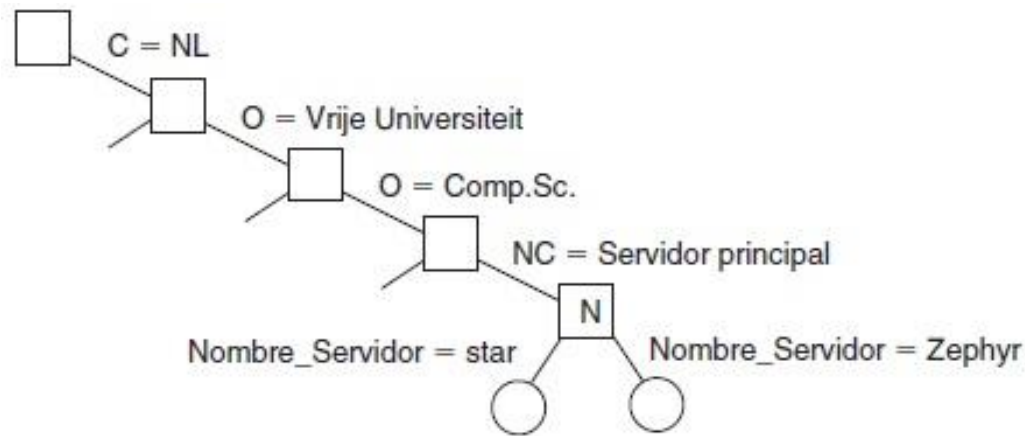
- Es un servicio levantado en el puerto 389 que se accede a través de un URL o FQDN ... ejm

ldap://dominio:port/ ... DN? ...?attributes ...scope? ... filters? ... extern? ...

Atributo	Abreviatura	Valor
País	P	NL
Localidad	L	Amsterdam
Organización	O	Vrije Universiteit
UnidadOrganizacional	UO	Comp.Sc.
NombreComún	NC	Main server
Servidores_Correo	2	137.37.20.3, 130.37.24.6, 137.37.20.10
Servidor_FTP	2	130.37.20.20
Servidor_WWW	2	130.37.20.20

# Características de LDAP

- Normalmente se organiza jerárquicamente donde un nodo es un registro o un directorio. El protocolo ofrece primitivas como `read` para leer un registro o `list` para leer todos los hijos
- Por ejemplo supongamos una búsqueda de un servidor principal en el departamento de Computer Science en la Vrije Universiteit



# Búsqueda en LDAP

Atributo	Valor
País	NL
Localidad	Amsterdam
Organización	Vrije Universiteit
UnidadOrganizacional	Comp.Sc.
NombreComún	Servidor principal
Nombre_Servidor	Star
Dirección_Servidor	192.31.231.42

Atributo	Valor
País	NL
Localidad	Amsterdam
Organización	Vrije Universiteit
UnidadOrganizacional	Comp.Sc.
NombreComún	Servidor principal
Nombre_Servidor	Zephyr
Dirección_Servidor	137.37.20.10

- Normalmente el cliente se autentica y puede pedir una conexión vía SSL/TLS (puerto 636). Una vez dentro, dependiendo de sus permisos, puede agregar, modificar o borrar registros.
- El DN es un identificador único del registro que puede estar compuesto por varios atributos. Es muy relevante para la búsqueda
- Existe *Active Directory* de Microsoft u openLDAP que es un proyecto GNU

# LDAP con DHT

La clave está en aprovechar la jerarquía y los atributos para obtener identificadores que se usen para una DHT

```
descripción {  
  tipo = libro  
  descripción {  
    autor = Tolkien  
    título = LOTR  
  }  
  género = fantasía  
}
```



$h_1$  hash(tipo-libro)  
 $h_2$  hash(tipo-libro-autor)  
 $h_3$  hash(tipo-libro-autor-Tolkien)  
 $h_4$  hash(tipo-libro-título)  
 $h_5$  hash(tipo-libro-título-LOTR)  
 $h_6$  hash(tipo-genero-fantasia)

- El nodo responsable de  $h_1$  mantendrá la referencia al recurso real. Si varios nodos almacenan la referencia habrá tolerancia a fallos
- Con una consulta con  $h_1$ ,  $h_2$  y  $h_3$  se enviará a todos los nodos que informan de libros de Tolkien. Se deberían obviar los *hash* más generales ...