ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 20: Nomeação (parte 3)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

13/05/2024

Anteriormente...

- Nomeação plana: nomes arbitrários, sem semântica
- Nomeação estruturada: nome dividido em partes, dentro de um espaço de nomes (representado por uma árvore)

Porém, é preciso saber o nome da entidade

- Se as entidades tiverem uma conjunto de pares `(atributo, valor)`...

Nomeação baseada em atributos

Abordagem alternativa

Em muitos casos, seria mais conveniente nomear e procurar entidades pelos seus atributos $\Rightarrow$ serviços tradicionais de diretórios (ex: páginas amarelas).

Nomeação baseada em atributos

Abordagem alternativa

Em muitos casos, seria mais conveniente nomear e procurar entidades pelos seus atributos $\Rightarrow$ serviços tradicionais de diretórios (ex: páginas amarelas).

Pequeno problema:

Designar o conjunto de atributos disponíveis e seus possíveis valores (e fazer com que os usuários usem os valores de maneira consistente) não é trivial.

Nomeação baseada em atributos

Abordagem alternativa

Em muitos casos, seria mais conveniente nomear e procurar entidades pelos seus atributos $\Rightarrow$ serviços tradicionais de diretórios (ex: páginas amarelas).

Pequeno problema:

Designar o conjunto de atributos disponíveis e seus possíveis valores (e fazer com que os usuários usem os valores de maneira consistente) não é trivial.

Grande problema:

Operações de consulta pode ser muito caras, já que necessitam que os valores dos atributos procurados correspondam aos valores reais das entidades. Em princípio, teríamos que inspecionar todas as entidades => particularmente ruim se o sistema for distribuído.

Implementando serviços de diretório

Solução para busca escalável

Implementar um serviço de diretório básico como um banco de dados, e combiná-lo com um sistema de nomes estruturado tradicional.

Lightweight Directory Access Protocol (LDAP)

Cada entrada de diretório consiste um par `(atributo, valor)`, nomeada de forma única para facilitar buscas.

Exemplo de uma entrada de diretório:

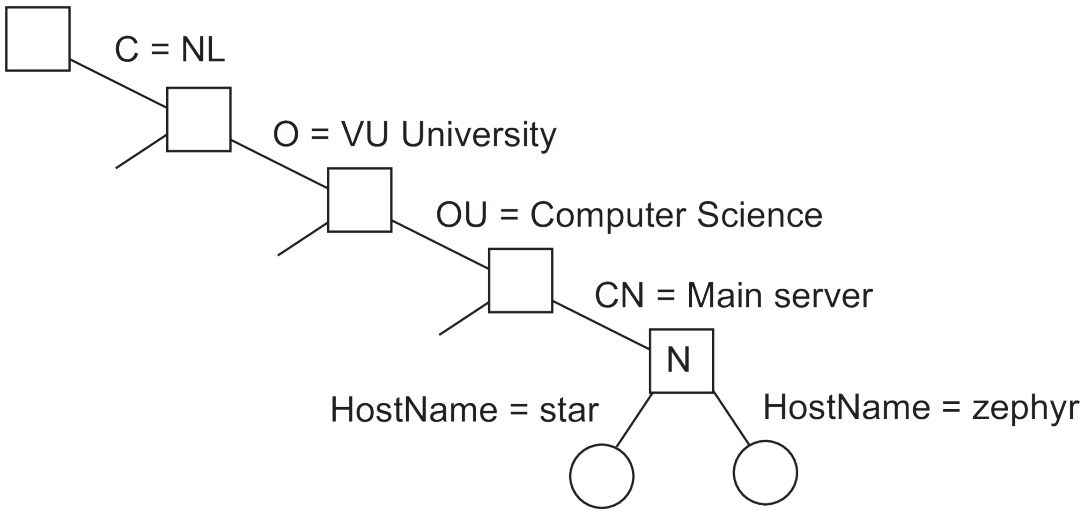
Attribute	**Abbr.**	**Value**
Country	*C*	NL
Locality	*L*	Amsterdam
Organization	*O*	VU University
OrganizationalUnit	*OU*	Computer Science
CommonName	*CN*	Main server
Mail_Servers	–	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	–	130.37.20.20
WWW_Server	–	130.37.20.20

LDAP

Essência

- Directory Information Base (DIB): coleção de todas as entradas de diretório de um serviço LDAP.
- Cada registro é nomeado de forma única como uma sequência de atributos nomeados (chamados de Relative Distinguished Name – RDN), de forma que possa ser buscado
- Directory Information Tree: o grafo que representa um serviço LDAP; cada nó representa uma entrada de diretório

Parte de uma árvore de diretório LDAP



LDAP

Exemplo de busca

Attribute	Value	Attribute	Value
Locality	*Amsterdam*	*Locality*	*Amsterdam*
Organization	*VU University*	*Organization*	*VU University*
OrganizationalUnit	*Computer Science*	*OrganizationalUnit*	*Computer Science*
CommonName	*Main server*	*CommonName*	*Main server*
HostName	*star*	*HostName*	*zephyr*
HostAddress	*192.31.231.42*	*HostAddress*	*137.37.20.10*

Resultado de `search("(C=NL)(O=VU University)(OU=Computer Science)(CN=Main server)")`

Distribuição pode ser obtida de forma semelhante ao DNS

- Operações com wildcards continuam custosas, e.g
  `search("(C=NL)(O=VU University)(OU=*)(CN=Main server)")`

Busca por atributos em sistemas P2P

- Sistemas P2P geralmente armazenam arquivos
- Sistemas P2P primitivos: é preciso conhecer a chave (hash) do arquivo
- Busca por atributos seria mais conveniente
- Uso de índices para tornar mais eficiente

Índice distribuído

Ideia geral

- Por hipótese, há um conjunto de atributos $\{a^1, \ldots, a^N\}$
- Cada atributo pode possuir valores a^k de um dado conjunto R^k
- Para cada atributo a^k, associe um conjunto de n_k servidores $\mathbf{S^k} = \{S_1^k, \ldots, S_{n_k}^k\}$
- Mapa global F: $F(a^k, v) = S_j^k$ com $S_j^k \in \mathbf{S^k}$ e $v \in R^k$

Observação

Se $L(a^k, v)$ é o conjunto de chaves de arquivos retornado por $F(a^k, v)$, então uma consulta pode ser formulada como uma expressão lógica, e.g.,

$$\big(F(a^1, v^1) \wedge F(a^2, v^2)\big) \vee F(a^3, v^3)$$

que pode ser processada no cliente construindo o conjunto

$$\big(L(a^1, v^1) \cap L(a^2, v^2)\big) \cup L(a^3, v^3)$$

Problemas com índices distribuídos

Vários

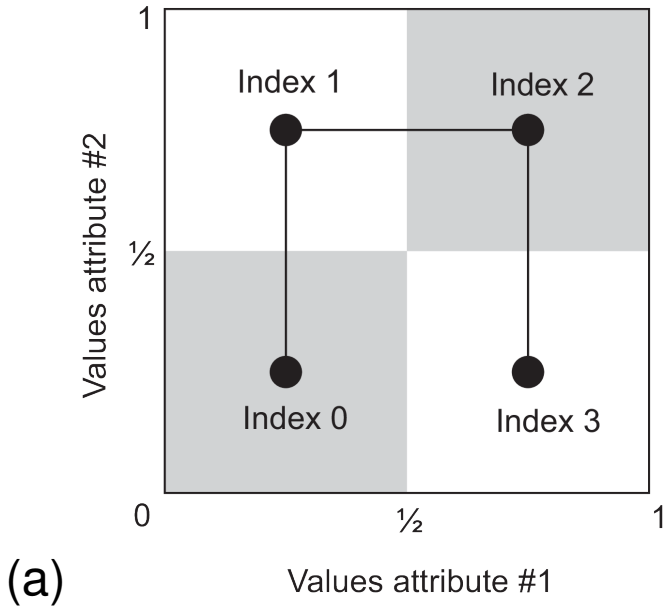
- Uma consulta envolvendo k atributos requer contato com k servidores
- Suponha uma consulta "$lastName = Silva \wedge firstName = Baltasar$": muitos registros deverão ser processados, pois há muitas pessoas com o sobrenome Silva.
- Não dá para implementar (facilmente) consultas com intervalos, como "$price = [1000 - 2500]$."

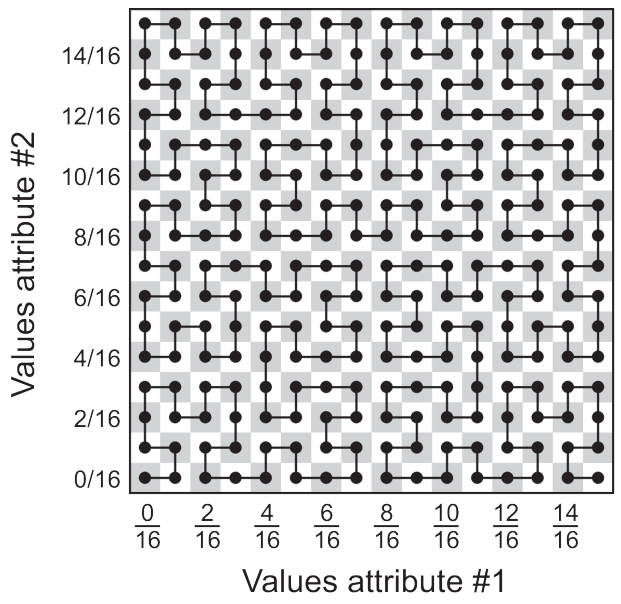
Alternativa: mapear atributos para um espaço unidimensional

Space-filling curves

1. Mapear o espaço N-dimensional definido pelos N atributos $\{a^1, \ldots, a^N\}$ para um espaço unidimensional
2. Preserva localidade
3. Usar hash para distribuir o espaço unidimensional entre os servidores de índice.

Hilbert space-filling curve de (a) ordem 1, e (b) ordem 4





Space-filling curve

Uma vez que curva seja definida

Considerando o caso de duas dimensões

- Uma curva de Hilbert de ordem k conecta 2^{2k} quadrantes $\Rightarrow$ tem 2^{2k} índices.
- Um intervalo de busca equivale a um retângulo R (no caso bidimensional)
- O retângulo R faz intersecção com um conjunto de quadrantes, cada um se referindo a um índice $\Rightarrow$ temos os indices necessários para fazer a busca

Acessando as entidades

Cada índice é mapeado para um servidor, que mantém uma referência para as entidades associadas. Possível solução: DHT

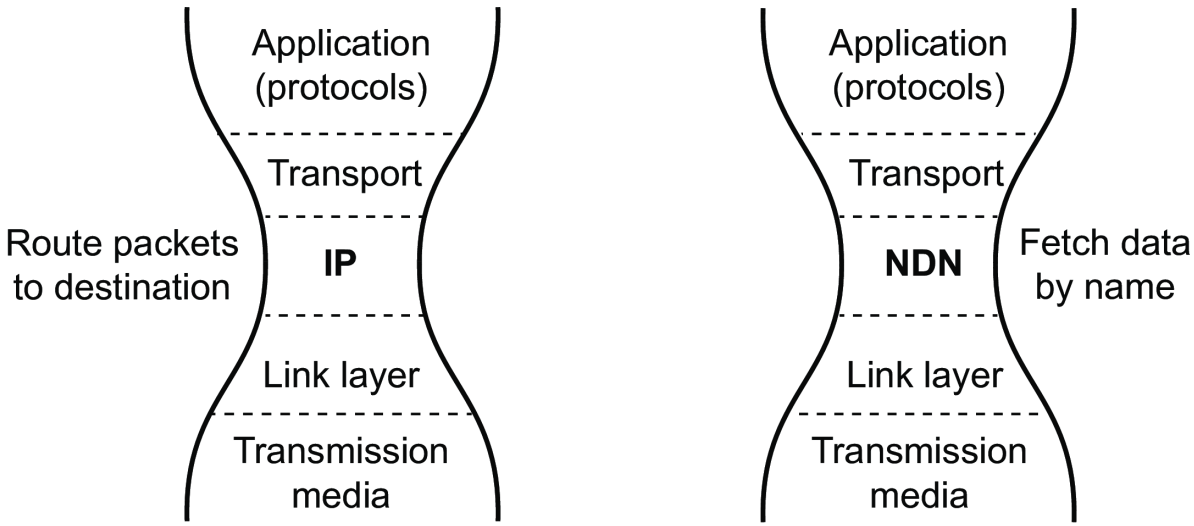
Named-data networking

Pontos principais

- Obter uma entidade através da rede utilizando seu nome, sem a necessidade de um endereço
- A rede deve usar o nome da entidade como entrada, roteando a requisição para a localidade da entidade
- NDN substitui o papel do IP numa arquitetura alternativa da internet

Exemplo de nome

/distributed−systems.net/books/Distributed Systems/4/01/Naming



Roteamento em NDN

Questão

Há alguma diferença fundamental entre rotear uma requisição como

distributed−systems.net/books/Distributed Systems/4/01/Naming

e rotear uma requisição para um endereço IPv6 como

2001:610:508:108:192:87:108:15 ?

Observação

Não há diferença fundamental.

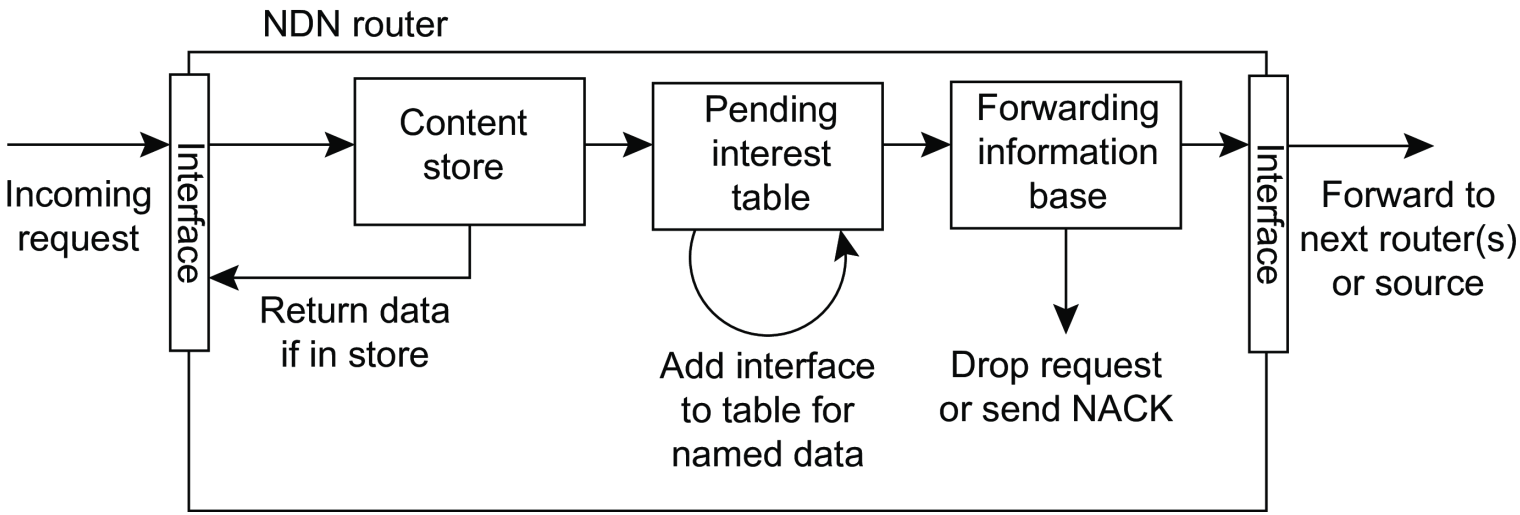
De uma forma ou de outra, uma parte do nome (normalmente um prefixo) é anunciado e usado para encaminhamento.

Roteamento em NDN

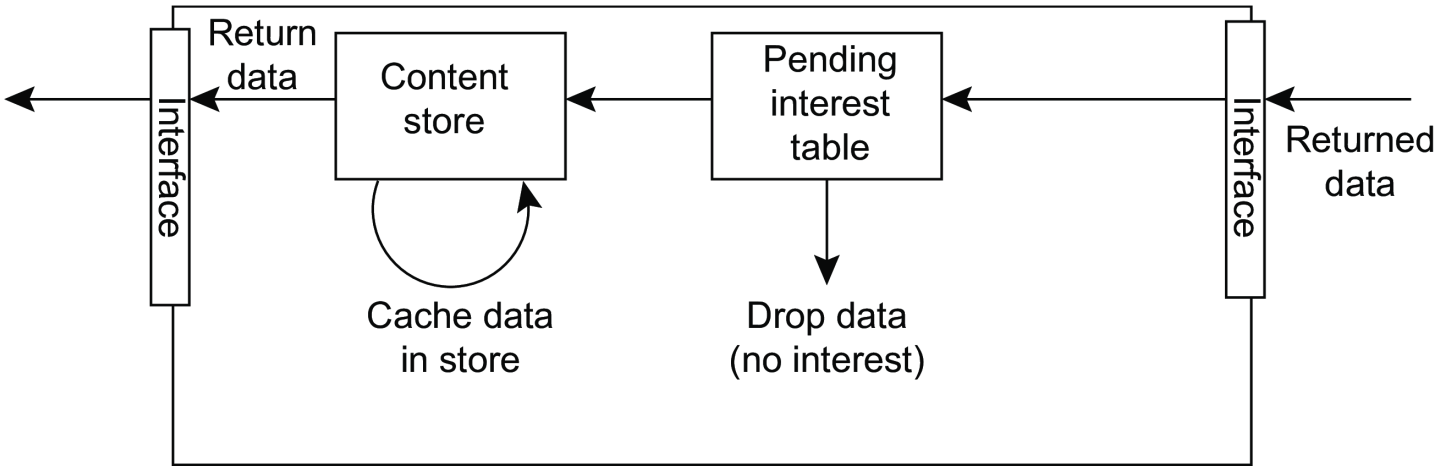
Elementos principais

- Content store: basicamente um cache para agilizar buscas já feitas anteriormente
- Pending interest table: tabela que associa interesses em dados às interfaces do roteador
- Forwarding information base: lida com o roteamento em si, e com a decisão do que fazer se um dado não estiver na base

Roteamento em NDN



Enviando uma requisição em direção ao seu destino destino



Retorno da resposta em direção ao requisitante