

Nomeação em Sistemas Distribuídos

Domain Name System (DNS)

Pedro Ximenes

Universidade Católica de Pernambuco (UNICAP)

March 15, 2026

- 1 Nomeação Estruturada
- 2 DNS e Camadas
- 3 Resolução de Nomes
- 4 DNS em Operação

Do identificador ao nome estruturado

Nomeação Estruturada: Motivação

Identificadores como IP, CPF ou códigos internos são precisos, mas nem sempre são amigáveis para memória humana.

Nomeação Estruturada: Ideia central

A **nomeação estruturada** organiza nomes em uma hierarquia, permitindo que pessoas localizem recursos por caminhos lógicos.

- Espaço de nomes modelado como **árvore**.
- Existe um **nó raiz** e nós internos.
- O recurso desejado costuma estar em um **nó folha**.

Exemplo 1: Sistema de arquivos

Caminho hierárquico

```
/home/user/unicap/arq-sistemas/aulas/aula11.pdf
```

- O caminho pode ser longo, mas cada nível acrescenta contexto.
- A estrutura facilita memorização e localização do arquivo final.
- A resolução ocorre percorrendo diretórios desde a raiz até a folha.

A hierarquia reduz ambiguidade: cada etapa do caminho restringe o conjunto possível de resultados.

Exemplo 2: Nomes de hosts na Internet

Exemplo de host

www.c3.unicap.br

- Leitura da direita para a esquerda: br $\rightarrow$ unicap $\rightarrow$ c3 $\rightarrow$ www.
- Cada rótulo representa um nível da hierarquia de nomeação.
- O formato torna o nome mais previsível e memorável.

DNS

O DNS **usa essa estrutura** para mapear nomes em endereços IP.

O que o DNS resolve?

Definição

O **Domain Name System (DNS)** realiza resolução de nomes: dado um nome legível, retorna o endereço IP do servidor responsável.

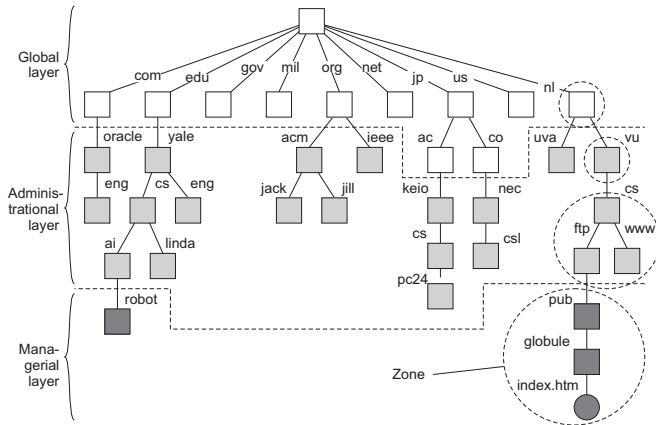
- Entrada: nome como `google.com.br`.
- Saída: endereço IP (ou conjunto de IPs) para contato com o serviço.
- Objetivo: desacoplar nome humano da localização de rede.

Consequência prática

O usuário lembra o nome; a infraestrutura encontra o endereço correto.

Camadas do espaço de nomes do DNS

- **Camada global:** porção superior da hierarquia (ex.: .br, .com, .org, .nl).
- **Camada administrativa:** organizações e instituições (ex.: unicap.br, uol.com.br, globo.com).
- **Camada gerencial:** partes mais próximas dos recursos, como serviços e caminhos de conteúdo, que mudam com maior frequência.



Comparando as camadas na prática

Escala e quantidade de nós

- Global: poucos nós, escopo mundial, altíssima responsabilidade.
- Administrativa: muitos nós, escopo por organização.
- Gerencial: volume muito alto de nomes, mudanças frequentes.

Desempenho esperado

- Global: pode tolerar respostas mais lentas, pois o cache ajuda bastante.
- Administrativa: resposta em milissegundos para manter boa experiência.
- Gerencial: resposta imediata para serviços e conteúdos ativos.

Issue	Global	Administrational	Managerial
Geographical scale	Worldwide	Organization	Department
Number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Client-side caching	Yes	Yes	Sometimes

Disponibilidade, réplicas e propagação

- Na camada global, falhas têm grande impacto, então **replicação** é essencial.
- Atualizações globais podem propagar lentamente devido ao volume e à necessidade de consistência entre réplicas.
- Na camada administrativa, atualizações precisam ser mais rápidas (ex.: ativação de conta, troca de serviço interno).
- Na camada gerencial, mudanças acontecem com alta frequência, exigindo controle de cache por tempo de vida (TTL).

Pergunta comum

Se a camada global pode ser lenta, por que um site famoso abre em menos de 1 s?

Resposta

Porque consultas anteriores ficam em cache no cliente e em resolvedores locais, evita-se repetir toda a travessia da hierarquia.

Problema de resolução

Objetivo

Resolver: `ftp://ftp.cs.vu.nl/pub/globe/index.html` até obter o IP do servidor.

- A busca parte de níveis superiores e desce na hierarquia.
- Em cada passo, um servidor informa quem conhece melhor o próximo nível.
- Duas estratégias clássicas: **iterativa** e **recursiva**.

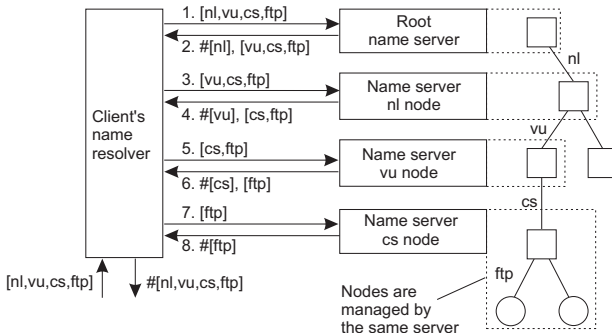
Resolução iterativa: ideia geral

Fluxo

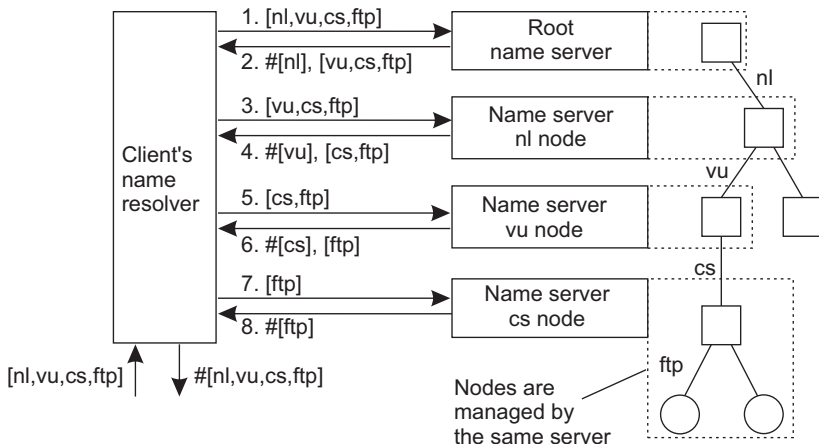
- 1 Cliente pergunta ao servidor raiz.
- 2 Raiz devolve referência para o próximo servidor (ex.: de .nl).
- 3 Cliente pergunta ao próximo servidor e repete o processo.
- 4 Ao final, cliente chega ao servidor autoritativo e obtém a resposta.

Característica principal

O cliente (ou resolvedor local) controla o passo a passo da travessia.



Resolução iterativa: ideia geral



Iterativa com cache no cliente

- Após resolver `ftp.cs.vu.nl`, o cliente guarda referências de servidores intermediários.
- Em uma nova consulta para `mail.cs.vu.nl`, pode **cortar caminho** e começar de um nível mais baixo.
- Isso reduz latência e carga nos servidores mais altos da hierarquia.

Efeito prático

A raiz DNS tende a ser consultada menos vezes para domínios acessados com frequência, pois o cache reaproveita respostas anteriores.

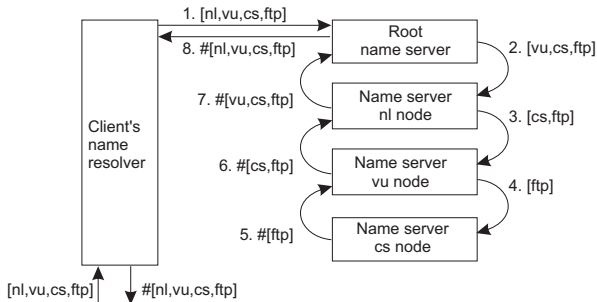
Resolução recursiva: ideia geral

Fluxo

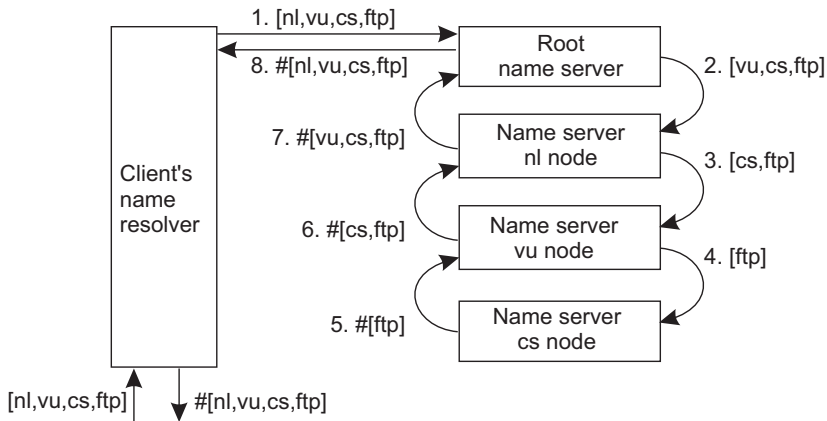
- 1 Cliente envia uma consulta ao servidor inicial.
- 2 Esse servidor repassa a consulta para o próximo, e assim por diante.
- 3 A resposta completa retorna pela cadeia de servidores até o cliente.

Característica principal

O cliente faz menos trabalho direto; os servidores assumem a coordenação.



Resolução recursiva: ideia geral



Recursiva: vantagens e limitações

Vantagens

- Menor carga de processamento no cliente.
- Melhor aproveitamento de cache ao longo dos servidores.
- Pode reduzir custo de comunicação em links de longa distância.

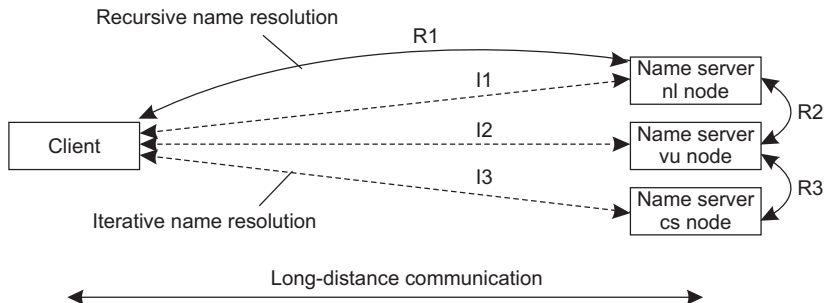
Limitações

- Maior carga e estado mantido nos servidores intermediários.
- Pode não ser adotada em servidores de topo devido ao custo.

Server for node	Should resolve	Looks up	Passes to child	Receives and caches	Returns to requester
cs	[ftp]	#[ftp]	—	—	#[ftp]
vu	[cs, ftp]	#[cs]	[ftp]	#[ftp]	#[cs] #[cs, ftp]
nl	[vu, cs, ftp]	#[vu]	[cs, ftp]	#[cs] #[cs, ftp]	#[vu] #[vu, cs] #[vu, cs, ftp]
root	[nl, vu, cs, ftp]	#[nl]	[vu, cs, ftp]	#[vu] #[vu, cs] #[vu, cs, ftp]	#[nl] #[nl, vu] #[nl, vu, cs] #[nl, vu, cs, ftp]

Iterativa vs. recursiva: custo de comunicação

- Em cenário iterativo, o cliente conversa separadamente com mais servidores de nomes.
- Em cenário recursivo, parte da comunicação acontece entre servidores que podem estar geograficamente mais próximos.
- Isso pode diminuir o custo total de tráfego no caminho cliente-servidor.



DNS em operação



Figure 1: Servidores raiz do DNS, que formam a base da hierarquia global: <https://root-servers.org/>

Comportamento observado na Web

- O DNS costuma operar com consultas leves e rápidas.
- Em muitos cenários, consultas DNS usam **UDP**, reduzindo overhead.
- Perdas de datagrama podem ocorrer; o cliente tenta novamente.
- Em navegação real, refresh e novas tentativas normalmente resolvem consultas perdidas.

Escalabilidade do DNS depende da combinação entre hierarquia, replicação estratégica e cache em múltiplas camadas.

Resumo da aula

O que vimos

- Nomeação estruturada como base para descoberta em larga escala.
- DNS como principal exemplo prático de sistema de nomes hierárquico.
- Camadas global, administrativa e gerencial e seus requisitos.
- Resolução iterativa e recursiva, com trade-offs de carga e desempenho.
- Papel central de cache para reduzir latência e tráfego.

O que aprendemos?

Nomes fornecem uma abstração estável para o usuário; o DNS transforma essa abstração em localização de rede de forma escalável.