

ACH 2147 — DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO DISTRIBUÍDOS

NOMES

Daniel Cordeiro

6 e 8 de junho de 2018

Escola de Artes, Ciências e Humanidades | EACH | USP

Greve(s) Aulas normais.

Greve(s) Aulas normais.

EP RPC. Enunciado no ar. Prazo: 6 de julho de 2018
(improrrogável). Vale 30% da média.

Greve(s) Aulas normais.

EP RPC. Enunciado no ar. Prazo: 6 de julho de 2018
(improrrogável). Vale 30% da média.

Prova 2 4 de julho (t.94) ~~27~~ e 29 de junho de 2018 (t.04)

Greve(s) Aulas normais.

EP RPC. Enunciado no ar. Prazo: 6 de julho de 2018
(improrrogável). Vale 30% da média.

Prova 2 4 de julho (t.94) ~~27~~ e 29 de junho de 2018 (t.04)

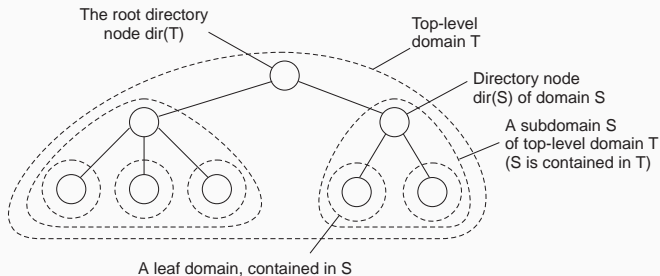
REC 20 de julho de 2018

- Nomes
 - identificadores;
 - espaço de nomes plano;
 - *finger tables* em DHTs

SERVIÇOS DE LOCALIZAÇÃO HIERÁRQUICOS (HLS)

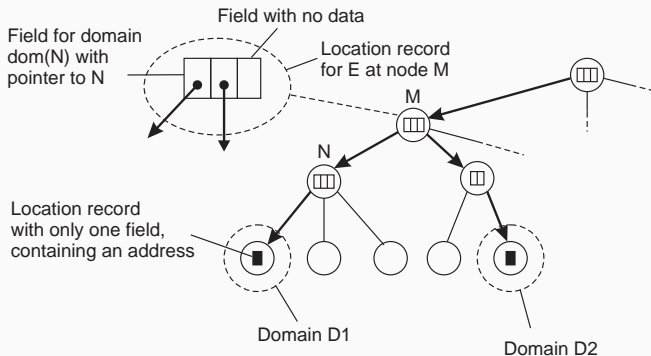
Ideia:

Construir uma árvore de busca em larga escala onde a rede é dividida em domínios hierárquicos. Cada domínio é representado por um diretório de nós separado.



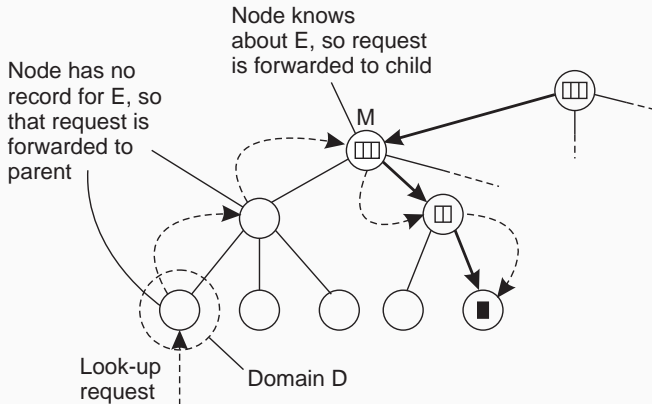
Invariantes

- Endereço de uma entidade E é armazenado numa folha ou nó intermediário
- Nós intermediários contêm um ponteiro para um filho sse a subárvore cuja raiz é o filho contenha o endereço da entidade
- A raiz conhece todas as entidades

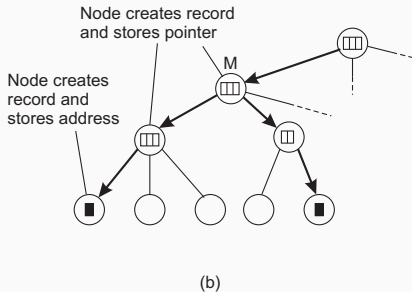
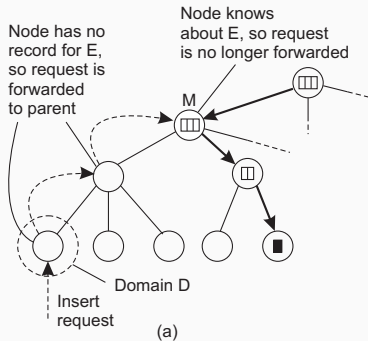


Princípios básicos:

- Comece a busca pelo nó folha local
- Se o nó souber sobre E , seguir o ponteiro pra baixo, senão continue a subir
- Continue subindo até encontrar a raiz



HLS: INSERÇÃO

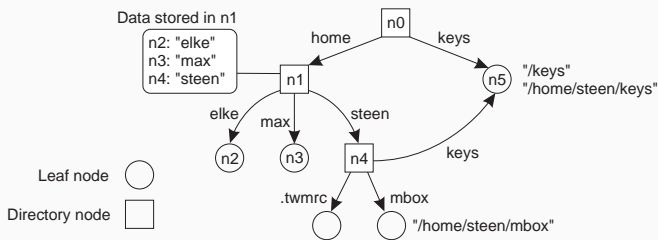


- (a) um pedido de inserção é encaminhado ao primeiro nó que conhece a entidade E
- (b) uma cadeia de ponteiros de redirecionamento é criada até o nó folha

ESPAÇOS DE NOMES ESTRUTURADOS

Ideia

Um grafo no qual um **nó folha** representa uma entidade que tem um nome. Um **diretório de nós** é uma entidade que se refere a outros nós.



Nota:

Um diretório contém uma tabela de pares (*identificador do nó*, *rótulo da aresta*)

Observação

É fácil guardar em um nó vários tipos de atributos para descrever propriedades da entidade que o nó representa:

- tipo da entidade
- identificador da entidade
- endereço da localização da entidade
- *nicknames*
- ...

Observação

É fácil guardar em um nó vários tipos de atributos para descrever propriedades da entidade que o nó representa:

- tipo da entidade
- identificador da entidade
- endereço da localização da entidade
- *nicknames*
- ...

Nota:

Nós de diretórios também podem ter atributos, além de guardar a tabela com os pares (identificador, rótulo)

Problema:

Para resolver um nome precisamos de um diretório. Como encontrar esse nó inicialmente?

Mecanismo de closure

- `each.usp.br`: inicia em um servidor DNS
- `/home/danielc/Maildir`: inicia em um servidor de arquivos NFS local (busca potencialmente recursiva)
- `00551130911033`: disca um número de telefone
- `200.144.248.41`: rota para um servidor da USP

Hard link

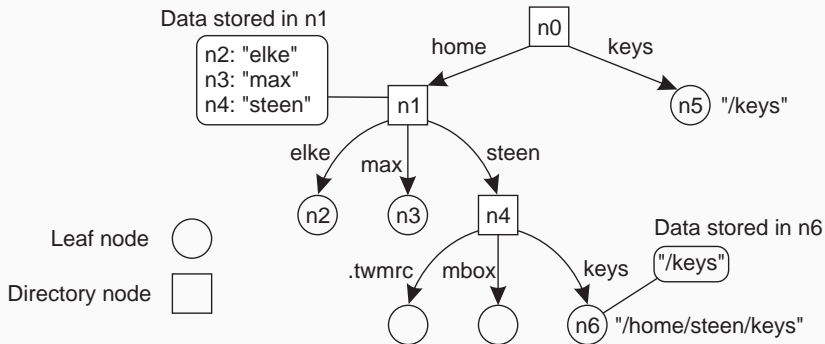
O que descrevemos até agora foi um **caminho**: um nome que é resolvido ao seguir um caminho específico em um grafo de nomes, indo de um nó para outro.

Soft link

Permite que um nó *O* mantenha o **nome** de outro nó:

- Primeiro resolva o nome de *O* (o que leva ao nó *O*)
- Leia o conteúdo de *O*, que leva ao **nome**
- A resolução do nome continua com **nome**

LINKS PARA NOMES



Observação:

O nó n5 tem apenas um nome.

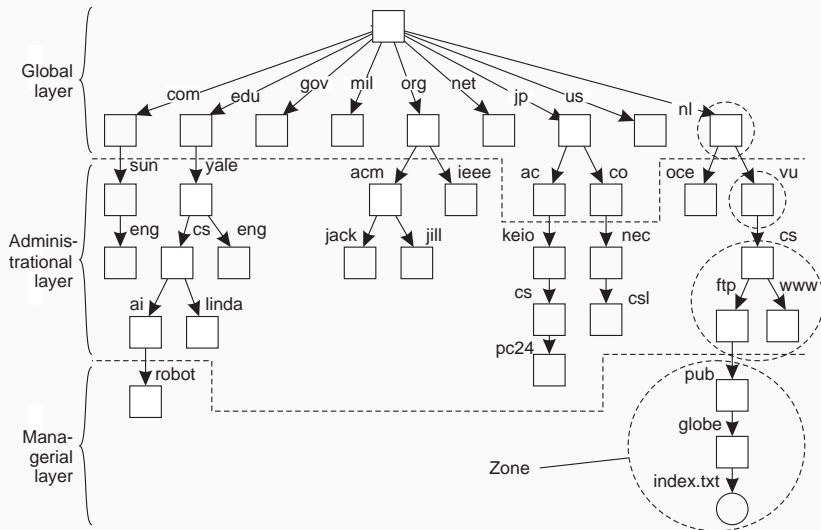
Problema:

Distribuir o processo de resolução de nomes, bem como o gerenciamento do espaço de nomes, em várias máquinas, de forma a distribuir o grafo de nomes.

Três níveis distintos:

1. **Nível global:** consiste de um diretório de nós de alto nível. Sua principal característica é que os nós do diretório devem ser gerenciado em conjunto por diferentes administradores
2. **Nível de administração:** nós de diretório de nível intermediário que podem ser agrupados de modo que cada grupo seja responsabilidade de um administrador diferente
3. **Nível gerencial:** nós de nível inferior que pertencem a um único administrador. O problema principal é mapear os nós de diretório aos servidores de nomes local.

IMPLEMENTAÇÃO DE ESPAÇOS DE NOMES

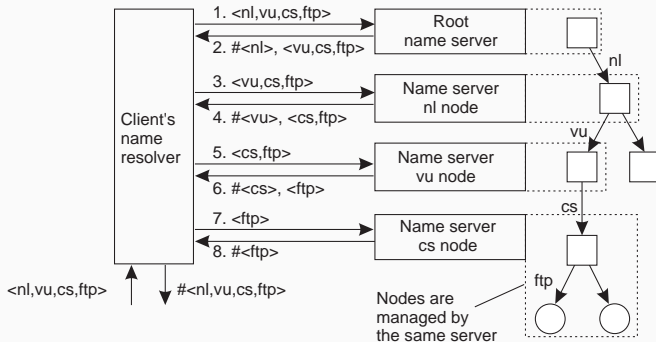


IMPLEMENTAÇÃO DE ESPAÇOS DE NOMES

Item	Global	Administração	Gerencial
1	Mundial	Organização	Departamento
2	Poucos	Muitos	Qdes. enormes
3	Segundos	Milissegundos	Imediato
4	Tardio	Imediato	Imediato
5	Muitos	Nenhum ou poucos	Nenhum
6	Sim	Sim	Às vezes
1: Escala geográfica 2: # Nós 3: Responsividade		4: Propagação de atualizações 5: # Réplicas 6: Cache no lado do cliente?	

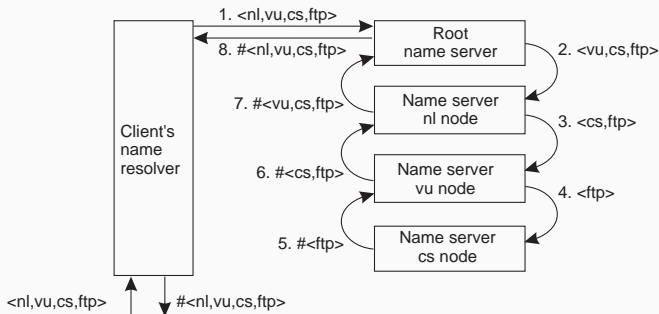
RESOLUÇÃO DE NOMES ITERATIVA

1. `resolve(dir,[name1,...,nameK])` enviado para **Server0** responsável por `dir`
2. **Server0** resolve `resolve(dir,name1) → dir1`, devolve a identificação (endereço) de **Server1**, que contém `dir1`.
3. Cliente envia `resolve(dir1,[name2,...,nameK])` para **Server1**, etc.



RESOLUÇÃO DE NOMES RECURSIVA

1. `resolve(dir,[name1,...,nameK])` enviado a `Server0`, responsável por `dir`
2. `Server0` resolve `resolve(dir,name1) → dir1` e envia `resolve(dir1,[name2,...,nameK])` para `Server1`, que contém `dir1`.
3. `Server0` espera pelo resultado de `Server1` e o devolve para o cliente.



USO DE CACHE NA RESOLUÇÃO RECURSIVA

Servidor para o nó	Deveria resolver	Procura por	Envia para filho	Recebe e and faz cache	Devolve ao solicitante
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>

Escalabilidade de tamanho

Devemos garantir que os servidores possam lidar com um grande número de requisições por unidade de tempo. Servidores de alto nível podem estar com problemas sérios.

Escalabilidade de tamanho

Devemos garantir que os servidores possam lidar com um grande número de requisições por unidade de tempo. Servidores de alto nível podem estar com problemas sérios.

Solução:

Assuma (pelo menos nos níveis global e de administração) que os conteúdos dos nós mudam muito pouco. Assim podemos aplicar replicação extensivamente, mapeando nós a múltiplos servidores e começar a resolução de nomes no servidor mais próximo.

Escalabilidade de tamanho

Devemos garantir que os servidores possam lidar com um grande número de requisições por unidade de tempo. Servidores de alto nível podem estar com problemas sérios.

Solução:

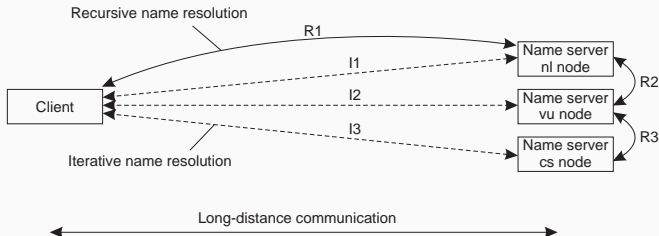
Assuma (pelo menos nos níveis global e de administração) que os conteúdos dos nós mudam muito pouco. Assim podemos aplicar replicação extensivamente, mapeando nós a múltiplos servidores e começar a resolução de nomes no servidor mais próximo.

Observação:

Um atributo importante em muitos nós é o seu endereço (onde a entidade pode ser localizada). Replicação faz com que servidores de nomes em larga escala sejam inadequados para localizar entidades móveis.

Escalabilidade geográfica

Precisamos garantir que a resolução de nomes escale mesmo em grandes distâncias geográficas



Porém:

Ao mapear nós em servidores que podem ser localizados em qualquer lugar, nós acabamos introduzindo uma dependência de localização implícita.

EXEMPLO: DNS DECENTRALIZADO

Ideia básica

Pegue um nome DNS completo, crie um hash com chave k e use um sistema DHT que permita consulta a chaves. **Desvantagem:** você não pode pedir por todos os nós em um subdomínio (mas pouca gente faz isso)

Informação em um nó:

SOA	Zone	Holds info on the represented zone
A	Host	IP addr. of host this node represents
MX	Domain	Mail server to handle mail for this node
SRV	Domain	Server handling a specific service
NS	Zone	Name server for the represented zone
CNAME	Node	Symbolic link
PTR	Host	Canonical name of a host
HINFO	Host	Info on this host
TXT	Any kind	Any info considered useful

Pastry

Sistema baseado em DHT que funciona com **prefixos** de chaves.

Considere um sistema onde as chaves pertençam a um espaço de números de 4 dígitos. Um nó com ID 3210 mantém informações sobre os seguintes nós:

n_k	prefix of $ID(n_k)$	n_k	prefix of $ID(n_k)$
n_0	0	n_1	1
n_2	2	n_{30}	30
n_{31}	31	n_{33}	33
n_{320}	320	n_{322}	322
n_{323}	323		

Notas:

O nó 3210 é responsável pelas chaves com prefixo 321. Se ele receber uma requisição pela chave 3012, ela será reencaminhada ao n_{30} . Para DNS: um nó responsável pela chave k armazena os registro DNS de nomes cujo valor do hash seja k .

Definição

Replicação no nível i – registro é replicado em todos os nós com prefixo i . **Nota:** # saltos para procurar por um registro no nível i normalmente é i .

Observação:

Seja x_i a fração dos nomes DNS mais populares que deveriam ser replicados no nível i . Então:

$$x_i = \left[\frac{d^i (\log N - C)}{1 + d + \dots + d^{\log N - 1}} \right]^{1/(1-\alpha)}$$

onde N é o número total de nós, $d = b^{(1-\alpha)/\alpha}$ e $\alpha \approx 1$, assumindo que a “popularidade” siga a distribuição de Zipf: a frequência do n -ésimo item mais popular é proporcional a $1/n^\alpha$

Se quisermos que o número médio de consultas para a resolução de um nome DNS seja $C = 1$, então com $b = 4$, $\alpha = 0,9$, $N = 10.000$ e 1.000.000 registros

61 registros mais populares devem ser replicados no nível 0

284 próximos registros mais populares no nível 1

1323 próximos registros mais populares no nível 2

6177 próximos registros mais populares no nível 3

28826 próximos registros mais populares no nível 4

134505 próximos registros mais populares no nível 5

o resto não deve ser replicado

Observação

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

Observação

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

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.

Observação

Em muitos casos, é mais conveniente nomear e procurar entidades pelos seus **atributos** ⇒ serviços tradicionais de diretórios (ex: páginas amarelas).

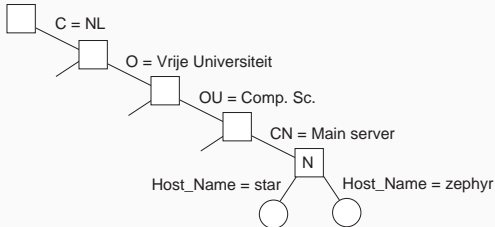
Problema:

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.

Solução:

Implementar serviços de diretórios básicos (tais como bancos de dados) e combiná-los com os sistemas de nomes estruturados tradicionais.

EXEMPLO: LDAP



Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	137.37.20.10

```
answer = search("&(C = NL)  
              (O = Vrije Universiteit)  
              (OU = *)  
              (CN = Main server)")
```