

SISTEMAS DISTRIBUÍDOS

Capítulo 10

Introdução aos sistemas de designação, de descoberta e de localização de serviços

NOTA PRÉVIA

A apresentação utiliza algumas das figuras livro de base do curso

G. Coulouris, J. Dollimore and T. Kindberg,
Distributed Systems - Concepts and Design,
Addison-Wesley, 4th Edition, 2005

DA NECESSIDADE DOS NOMES

Num sistema distribuído os nomes são imprescindíveis para designar computadores, serviços, utilizadores, objectos remotos, ficheiros e recursos em geral, ... Os diferentes componentes do sistema, assim como os utilizadores, só podem partilhar recursos se os poderem designar.

Os nomes permitem designar e identificar entidades ou objectos.

Por vezes, os nomes incluem informação relativa a propriedades/atributos dos objectos:

Exemplos?

"xpto@foo.com", <http://asc.di.fct.unl.pt/sd1>

Um **serviço de nomes** permite obter dados (atributos) sobre uma entidade dado o seu nome.

Um **serviço de directório** ou **descoberta** permite obter dados sobre as entidades que satisfazem uma dada descrição.

NOMES E IDENTIFICADORES

Designar - acção de apontar, indicar, mostrar, escolher

Identificar - tornar idêntico, o que faz com que uma coisa seja idêntica a outra, o que permite saber se duas coisas são distintas ou não

Um **nome** permite designar uma entidade

Num dado contexto, uma entidade pode ser designada por mais do que um nome.

Um **identificador** permite identificar uma entidade

Num dado contexto, uma entidade tem um e um só identificador.

Dois identificadores distintos identificam duas entidades diferentes. Se duas entidades tiverem o mesmo identificador, então são a mesma entidade.

Um **identificador único** permite identificar uma entidade de forma permanente

Se $UID1=UID2$, então $Ent1=Ent2$ para todo o sempre

Se $UID1 \neq UID2$, então $Ent1 \neq Ent2$ para todo o sempre

NOMES, IDENTIFICADORES E ENDEREÇOS NA PRÁTICA

Utiliza-se o termo **nome simbólico**, textual, orientado aos utilizadores, ou externo para designar ou identificar entidades através de nomes mnemónicos e legíveis para os utilizadores. Exemplos?

`www.di.fct.unl.pt, /home/nmp/myfile`

Utiliza-se o termo **identificador único sistema (UID)** para designar sequências de símbolos, geralmente sem significado mnemónico, que permitem identificar entidades de forma (quase) permanente, ao nível interno do sistema distribuído. Exemplos?

`AF.65.8F.89.1B.23.FF.45.A5.89.8B` (uid de um objecto remoto)

Utiliza-se o termo **endereço** para designar formas especiais de designação transitória, volátil ou temporária das entidades, geralmente associadas à localização das mesmas. Exemplo?

`10.0.0.12`

Um endereço é um nome que dá acesso imediato a uma entidade.

URLs, URNs, URIs

Vantagens dos URNs?

Vantagens dos URLs?

Um **URL** (Uniform Resource Locator) é um nome que indica a localização de um objecto

scheme:scheme-specific-location

Ex: `http://asc.di.fct.unl.pt/sd1`

Um URL é basicamente um endereço

Um **URN** (Uniform Resource Name) é um nome que permite identificar um objecto independentemente da sua localização

urn:nameSpace:nameSpace-specificName

Ex: `urn:ISBN:0-201-64233-8`

Para aceder a um objecto será geralmente necessário existir um serviço de nomes que converta um URN num URL

URLs e URNs são URIs (Uniform Resource Identifiers)

GERAÇÃO DE UUIDs

De forma geral os UUIDs asseguram coerência referencial e propriedades dos identificadores devido a não serem reutilizados. Esta propriedade torna a sua geração mais difícil.

A forma mais fácil de se gerarem UUIDs consiste em utilizar geradores de números aleatórios ou estampilhas horárias.

Os UUIDs não são geralmente visíveis aos utilizadores ou são complementados com "sugestões" ("hints"). Alternativamente usam-se nomes não puros, prefixando-os com endereços de sites, por exemplo.

NOMES GLOBAIS *VERSUS* NOMES CONTEXTUAIS

Todos os nomes são contextuais pelo que duas entidades computacionais só podem partilhar um nome se partilharem directa ou indirectamente um contexto comum de interpretação de nomes. Caso contrário não se “entenderiam”.

Um sistema distribuído necessita de um contexto comum a todas as entidades computacionais, para que essas entidades possam interagir. Os nomes relativos a esse contexto global dizem-se **nomes globais**, por oposição a nomes relativos a outros contextos mais limitados que se dizem **nomes contextuais**.

Os nomes globais são independentes do contexto e podem passar-se livremente entre as diferentes entidades computacionais.

EXEMPLOS

Nomes globais:

nfs://phoenix.students.di.fct.unl.pt/home/joe

<http://www.google.com>

smb://fatdata.berkley.edu/students/johnDeere

rmi://bigserver.di.fct.unl.pt/computeService

193.136.122.1

Nomes contextuais:

10.200.0.2

/home/joe

C:\database\students

ORGANIZAÇÃO DO ESPAÇO DE NOMES

Um **domínio de nomeação** é um espaço de nomes para o qual existe uma única autoridade administrativa para atribuir nomes.

Em geral, num sistema coexistem vários contextos e domínios de designação e a tradução dos nomes envolve interpretações parciais em domínios do mesmo nível ou de níveis diferentes.

Assim, estabelecem-se “ligações” entre os diferentes domínios e procedem-se a várias traduções de nomes até se chegar ao objecto. Exemplo:
<http://asc.di.fct.unl.pt/sd1/aulas-teoricas/cap6.pdf>

Quais os vários domínios ?

Geralmente, os contextos iniciais são globais e são materializados por um serviço de nomes. Os outros contextos podem ser materializados por outros servidores de nomes ou directamente pelos servidores que gerem os objectos.

SERVIÇO DE DESIGNAÇÃO OU DE NOMES

Um serviço de designação ou **serviço de nomes** é um serviço que permite obter um conjunto de atributos de uma entidade dado o seu nome. A resolução do nome é realizada a partir de um contexto de interpretação do nome.

Cada um desses nomes pode designar utilizadores, servidores, serviços, objectos remotos, ficheiros, etc.

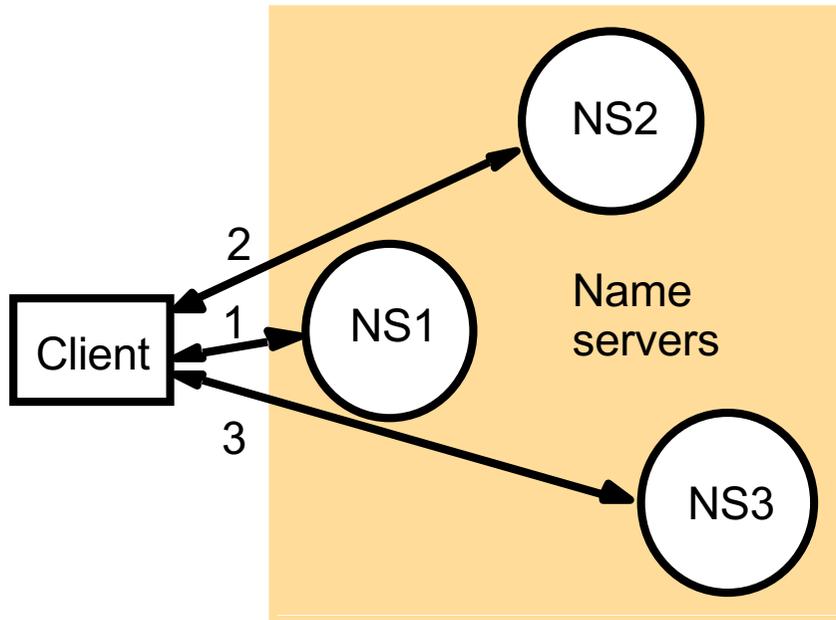
Um **serviço de nomes** geralmente implementa uma base de dados (distribuída) que associa nomes aos atributos das entidades que estes designam.

Existem duas operações importantes:

Lookup: dado um nome devolve os atributos associados ao mesmo

Bind: que associa um nome a um conjunto de atributos

RESOLUÇÃO ITERATIVA DO NOME PELO CLIENTE



Características?

Servidor processa cada pedido rapidamente.

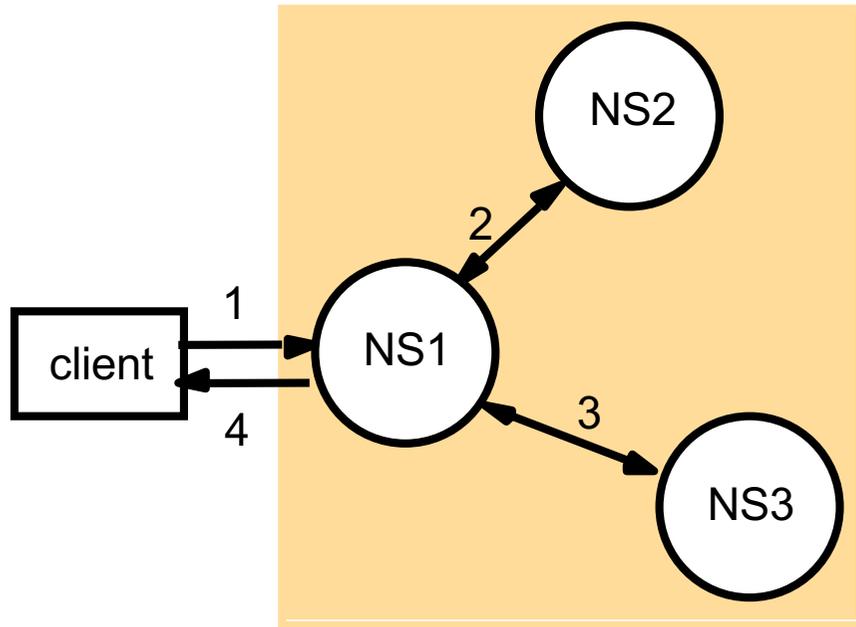
Resolução começa em servidores diferentes.

Cliente apresenta nome ao servidor local de nomes

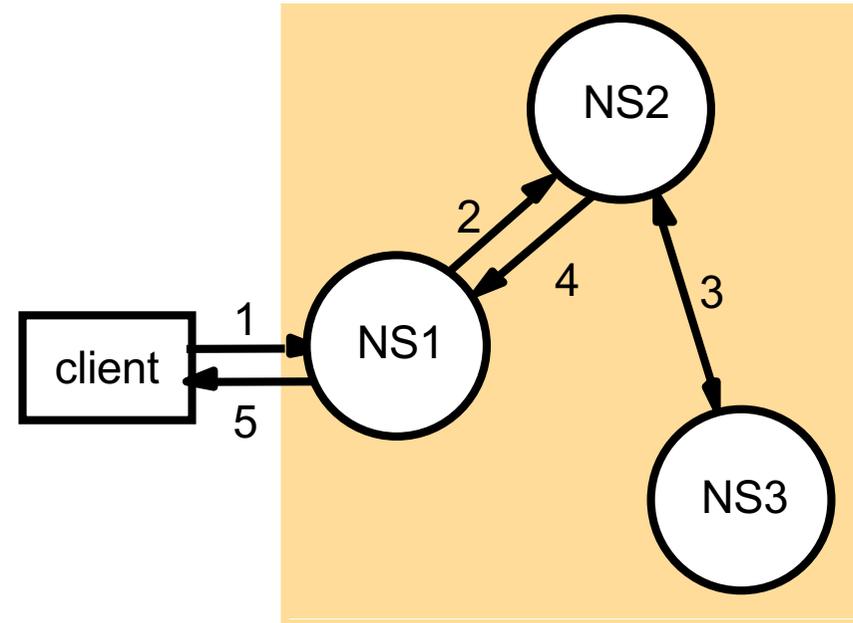
Se servidor conhece o nome, devolve atributos

Se servidor não conhece o nome, indica outro servidor onde resolver o nome

RESOLUÇÃO RECURSIVA DO NOME PELOS SERVIDORES



Non-recursive
server-controlled



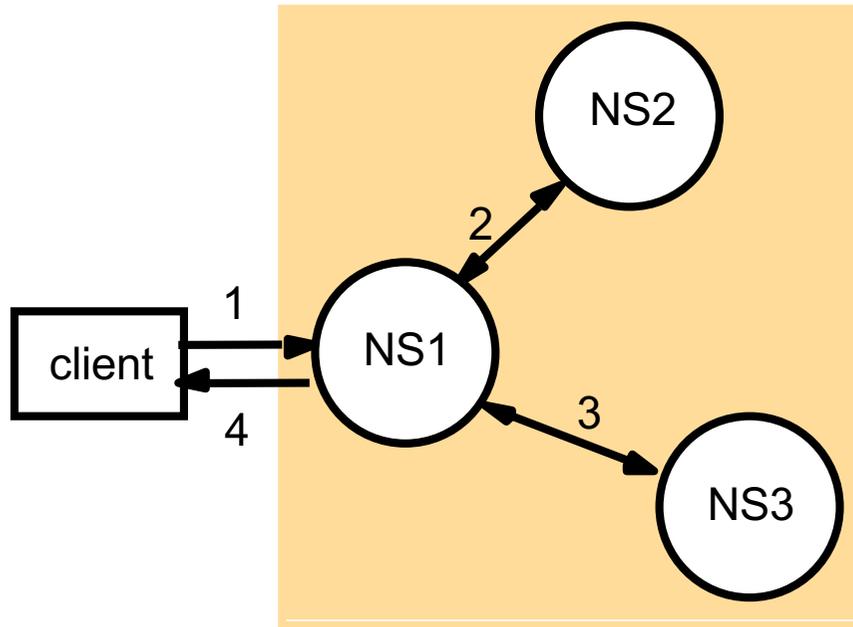
Recursive
server-controlled

Cliente apresenta nome ao servidor local de nomes

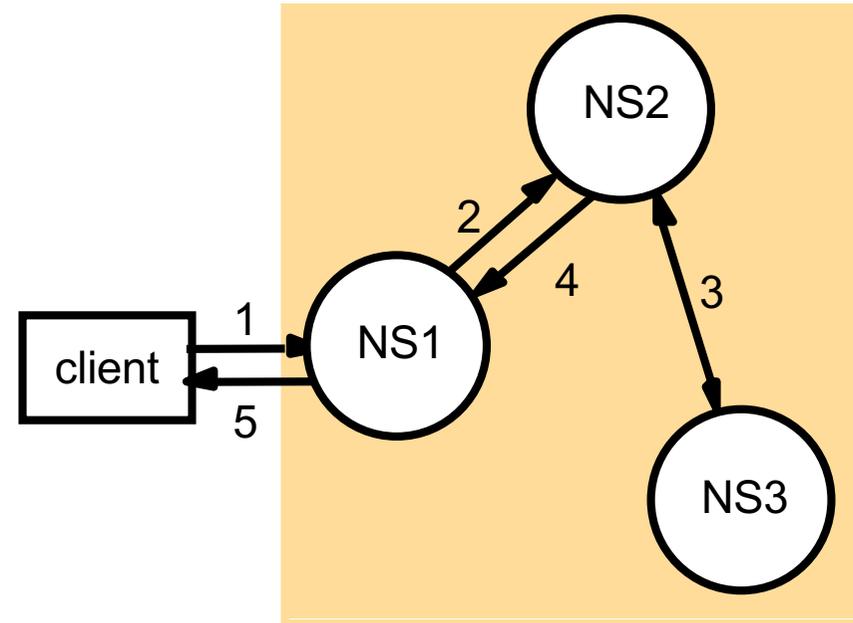
Se servidor conhece o nome, devolve atributos

Se servidor não conhece o nome, o próprio servidor resolve o nome iterativamente ou recursivamente

RESOLUÇÃO RECURSIVA DO NOME PELOS SERVIDORES



Non-recursive
server-controlled



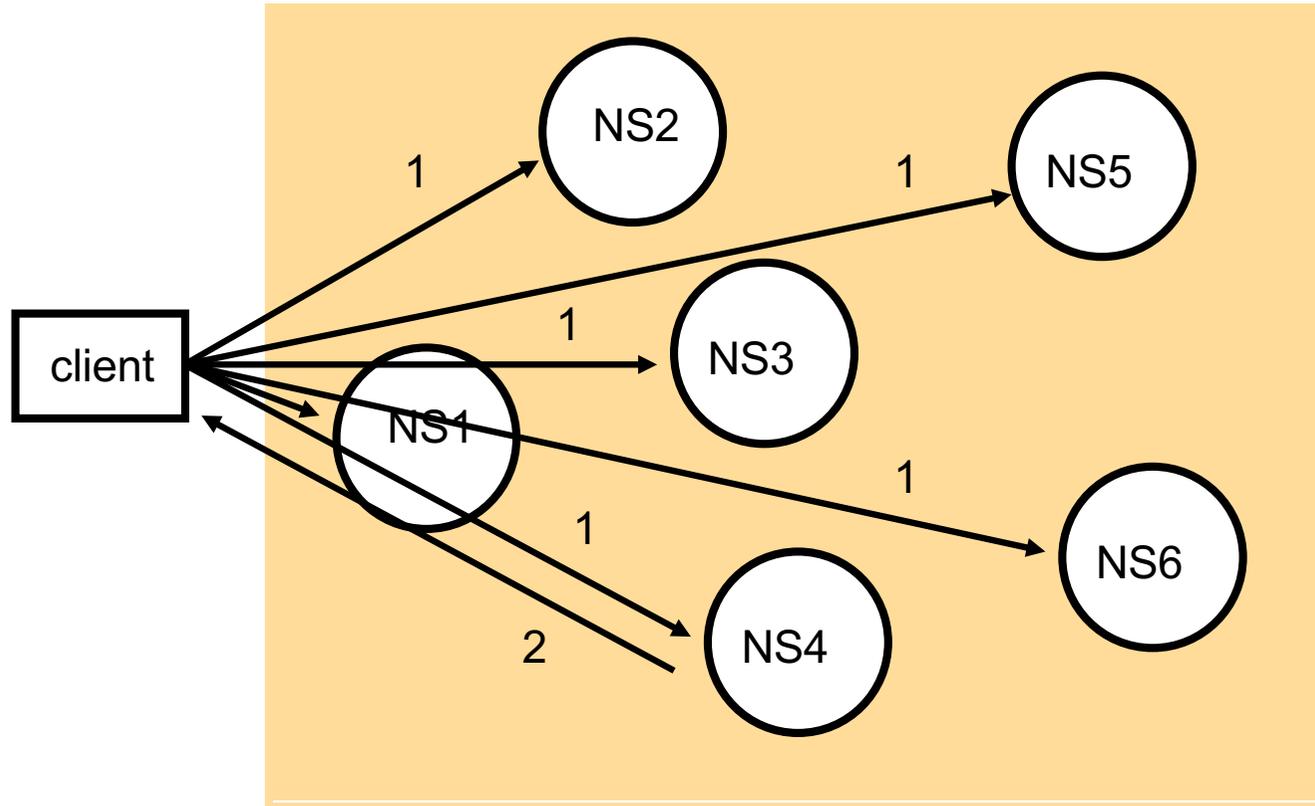
Recursive

Cliente apresenta nome ao servidor local
Se servidor conhece o nome, devolve resposta
Se servidor não conhece o nome, o servidor consulta recursivamente

Características?

- Cada pedido fica pendente até estar resolvido (ocupando recursos).
- Pode permitir contornar problemas de permissões.
- Resolução começa em servidores diferente.

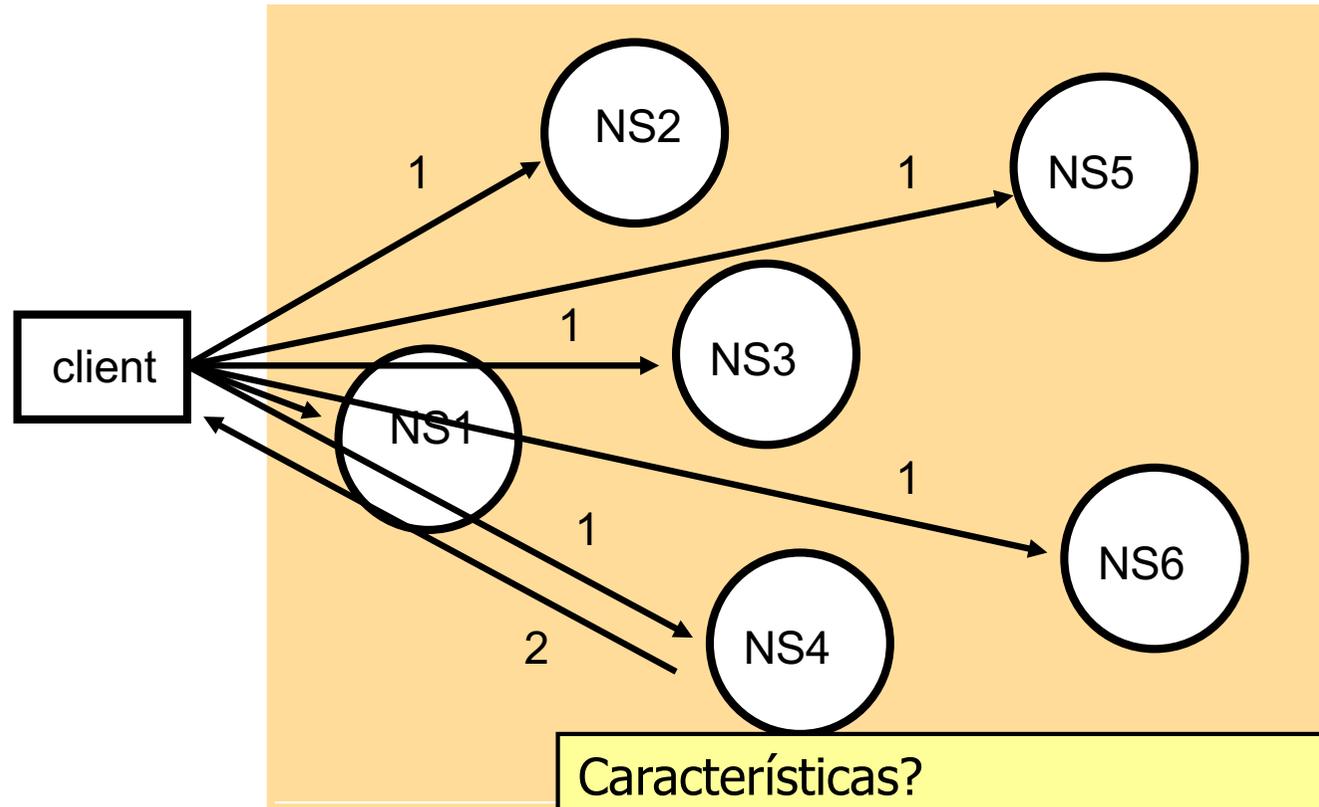
RESOLUÇÃO POR MULTICASTING



Cliente envia mensagem *multicast* aos servidores de nomes

O servidor de nomes que conhece o nome, devolve os atributos

RESOLUÇÃO POR MULTICASTING



Cliente envia mensagem *multicast* aos
O servidor de nomes que conhece o

Características?

Apenas possível em ambientes que suportam *multicast*.
Todos os servidores de nomes vêem todos os pedidos.

MASCARAMENTO DAS FALHAS E *CACHING*

Serviço de nomes é muitas vezes fundamental no acesso ao sistema.

Para mascarar falhas e fornecer elevada disponibilidade pode recorrer-se à **replicação** e ***caching*** de contextos.

Ao colocar cópias da informação (réplicas) em servidores distintos é possível tolerar falhas dos servidores e da rede.

Como é que os clientes sabem quais são os servidores?

Problema idêntico a saber um servidor - por exemplo, os atributos de um contexto contém a lista das réplicas.

O ***caching*** e replicação também permitem melhorar a escalabilidade (e responder a um elevado número de pedidos).

Problema?

Manter coerência entre as réplicas e a informação "oficial".

SERVIÇOS DE NOMES: CONSISTÊNCIA FRACA

Muitos serviços de nomes **não asseguram coerência total**, isto é, permitem que os clientes observem incoerências momentâneas. O sistema só assegura que as incoerências não se vão manter e que todas as réplicas tendem para um estado coerente.

Propriedades geralmente assumidas:

- os clientes suportam algum grau de incoerência.
- os valores registados no serviço evoluem lentamente.

São frequentes as seguintes soluções:

- Replicação do tipo primário/secundários com propagação assíncrona das actualizações do primário para o secundário;
- Mecanismos de *caching* em que os valores são retirados da cache por um mecanismo de “envelhecimento” ou TTL, controlado pelos administradores dos contextos de designação. Trata-se de um mecanismo “optimista”.

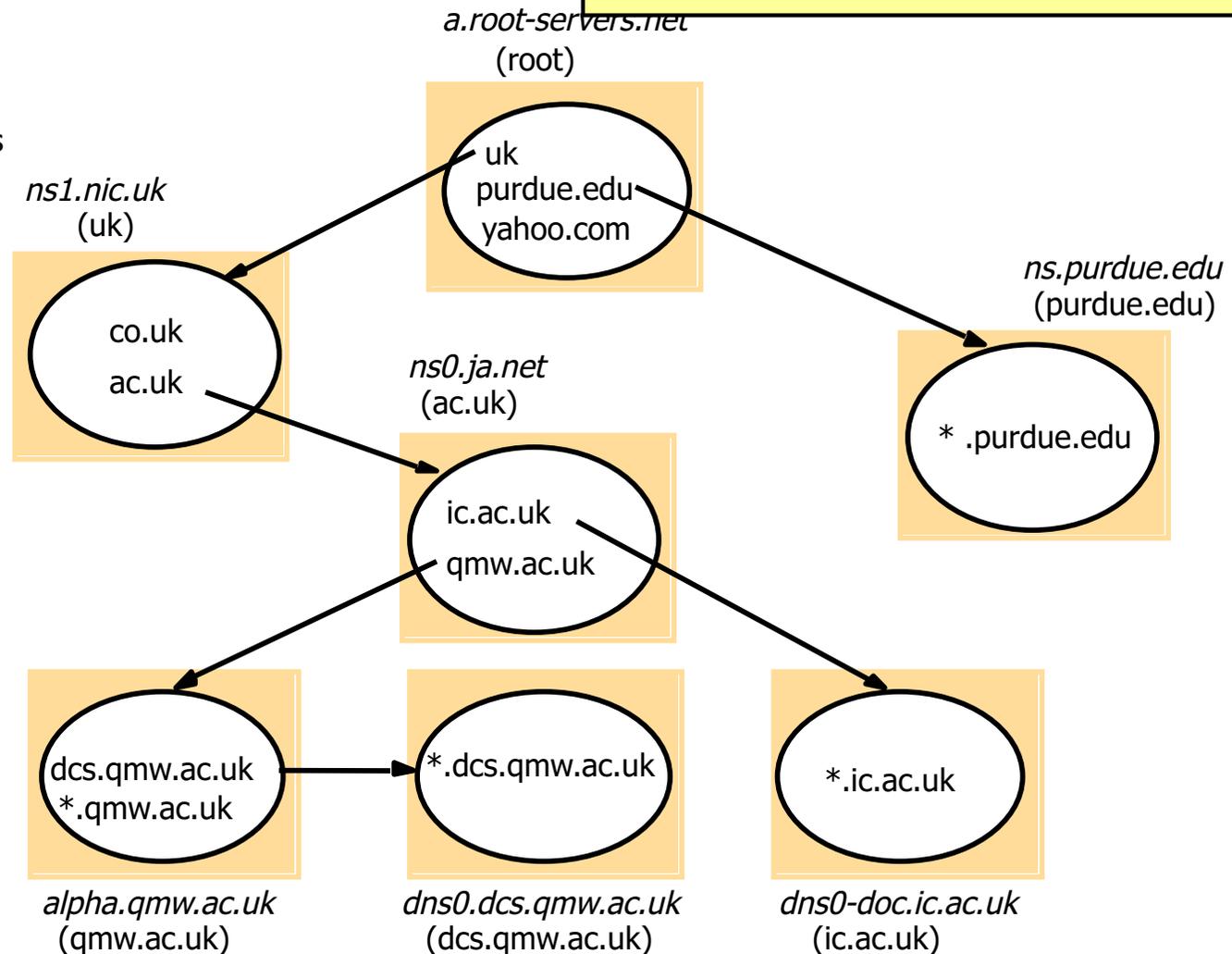
EXEMPLO DO DNS

DNS:

Sistema que mantém base de dados distribuída com informação de máquinas na internet.
Protocolo de acesso ao sistema.

Note: Name server names are in italics, and the corresponding domains are in parentheses.

Arrows denote name server entries



DOMÍNIOS DE TOPO

Domínios genéricos: com, edu, org, mil, etc.

Domínios nacionais: pt, uk, etc.



REGISTOS DNS (DNS RESOURCE RECORDS)

DNS: db distribuída com registos (RR)

Formato de um RR: (name, type, value, ttl)

- Type=A
 - O nome é um hostname
 - O valor é um endereço IP do host
- Type=NS
 - O nome é um domínio (e.g. foo.com)
 - O valor é o hostname de um servidor do domínio
- Type=CNAME
 - O nome é um alias para o nome "canónico" (o nome real)
 - O valor é o nome canónico
- Type=MX
 - O valor é o nome de um mail server do domínio e a respectiva prioridade

ARQUITECTURA DNS

Base de dados particionada por múltiplos servidores

Organização hierárquica dos servidores

Conjunto de servidores replicam informação da raiz da árvore

ARQUITECTURA DNS

Base de dados particionada por múltiplos servidores

Organização hierárquica dos servidores

Conjunto de servidores replicam informação da raiz da árvore

Informação sobre cada zona mantida em pelo menos dois servidores (de forma authoritative – em princípio actual)

Replicação primary/backup: backup lê informação periodicamente do primário

Frequência da verificação é parâmetro da configuração

Qualquer servidor pode fazer cache de informação de outros servidores

Entradas têm time-to-live

Clientes contactam servidores (pode ser dada lista de servidores)

Protocolo tipicamente em UDP

Cliente pode solicitar navegação recursiva ou iterativa. Servidor é livre de respeitar pedido.

Clientes podem fazer caching dos resultados

Esta arquitectura garante coerência da informação?

Porque funciona bem?

DNS: QUE FUTURO ?



Insights from Googlers into our products, technology, and the Google culture.

Introducing Google Public DNS

12/03/2009 08:35:00 AM

When you type www.wikipedia.org into your browser's address bar, you expect nothing less than to be taken to Wikipedia. Chances are you're not giving much thought to the work being done in the background by the [Domain Name System](#), or DNS.

Today, as part of our [ongoing effort to make the web faster](#), we're launching our own public DNS resolver called [Google Public DNS](#), and we invite you to try it out.

Most of us aren't familiar with DNS because it's often handled automatically by our Internet Service Provider (ISP), but it provides an essential function for the web. You could think of it as the switchboard of the Internet, converting easy-to-remember domain names — e.g., www.google.com — into the unique Internet Protocol (IP) numbers — e.g., 74.125.45.100 — that computers use to communicate with one another.

The average Internet user ends up performing hundreds of DNS lookups each day, and some complex pages require multiple DNS lookups before they start loading. This can slow down the browsing experience. Our research has shown that [speed matters](#) to Internet users, so over the past several months our engineers have been working to make improvements to our public DNS resolver to make users' web-surfing experiences faster, safer and more reliable. You can read about the specific technical improvements we've made in our product documentation and get installation instructions from our [product website](#).

SERVIÇOS DE NOMES: CONSISTÊNCIA FORTE

Muitas vezes é importante ter informação sobre nomes fortemente consistente

E.g.: quem é o servidor que é o primário

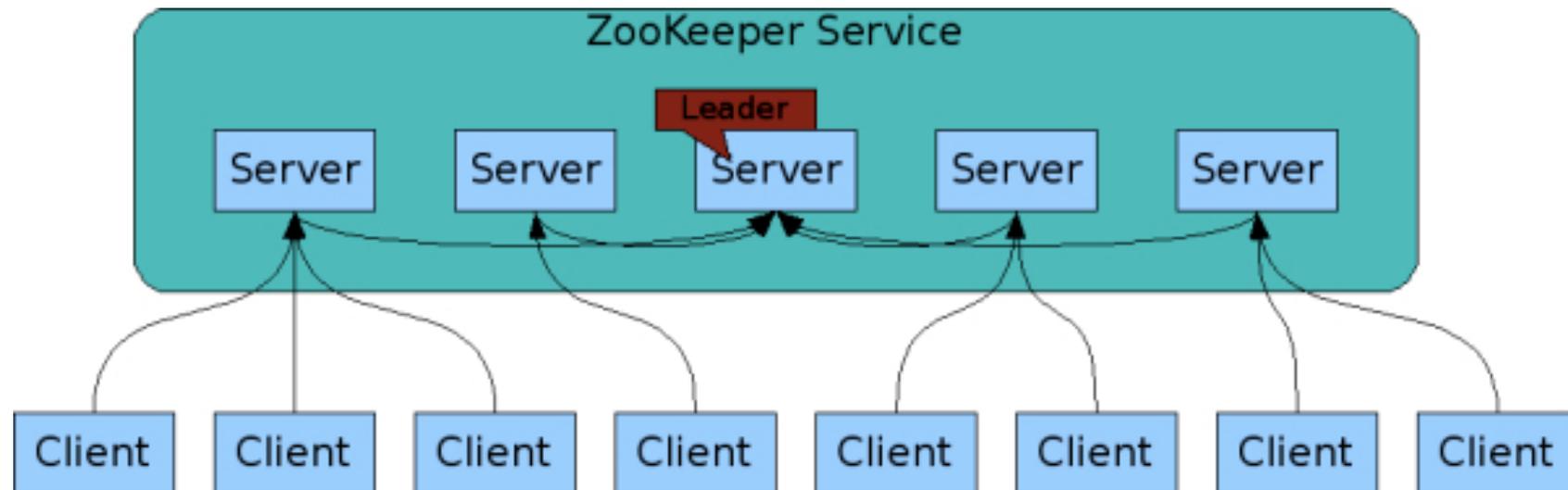
Serviço de nomes replicado, mas fortemente consistente

E.g. Zookeeper

ZOOKEEPER: ARQUITETURA

Conjunto de servidores mantém réplicas da base de dados

Operações totalmente ordenadas



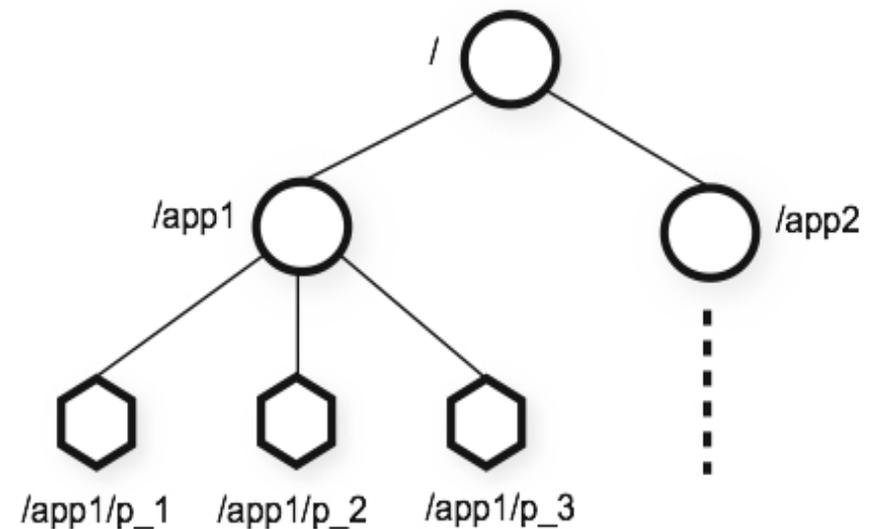
ZOOKEEPER: MODELO DE DADOS

Nós organizados numa estrutura hierárquica

“ZooKeeper was designed to store coordination data: status information, configuration, location information, etc., so the data stored at each node is usually small, in the byte to kilobyte range.”

Operações

Criar, remover, ler, escrever, ...
Escrever condicionalmente



DIRECTÓRIOS E SERVIÇOS DE DESCOBERTA

Um **serviço de directório** ou descoberta é um serviço que permite obter os atributos de uma entidade que satisfaz uma dada descrição (dada como um sub-conjunto de atributos).

Nestes sistemas, o nome é apenas um dos atributos.

Exemplo: qual é o endereço IP da impressora a cores do edifício II ?

DESIGNAÇÃO ATRAVÉS DE ATRIBUTOS

Um conjunto de atributos pode funcionar como “apontador” de uma entidade de uma forma mais potente que os nomes clássicos

Por exemplo, uma pessoa quando se dirige a uma agência de um banco pode dizer que deseja falar com o “responsável pela agência”, ou com o “caixa”.

Estes “apontadores” são abreviaturas de “a pessoa cujo atributo ‘função’ tem o valor ‘gerente’ ou ‘caixa’ ”.

Um mecanismo de designação baseado em atributos pode facilmente ser usado para designar conjuntos de entidades.

SERVIÇOS DE NOMES VS. SERVIÇOS DE DIRECTÓRIO

Serviços de nomes

- Nomes mais simples

Serviços de directório

- Atributos mais poderosos

 - Necessário definir atributos

- Mais simples obter serviços redundantes

- Mais simples para integração de um computador num *ambiente novo*

PARA SABER MAIS

G. Coulouris, J. Dollimore and T. Kindberg, Distributed Systems - Concepts and Design, Addison-Wesley, 4th Edition, 2005

Capítulo 9.